COMMODORE Disk User

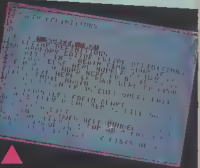
NOVEMBER/DECEMBER 1988
£2.50

Going for Warp Speed

Disk commands

We visit the Disk Doctor

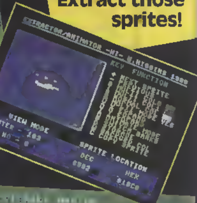
FOR C64 AND C128 USERS



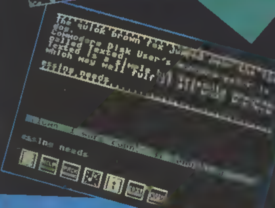
▲ CDU goes FORTH

Foxx Fights Back reviewed

Extract those sprites!

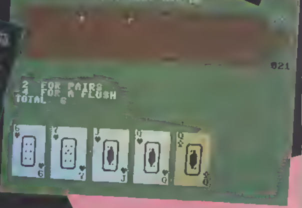


EXTRACTOR/ANIMATOR -M1- W.HIGGINS 1988
KEY FUNCTION
VIEW MODE
POINTER - 103
BANK NO - 0
SPRITE LOCATION
DEC 6592
HEX $19C0



The quick brown fox jumps
dog.
Commodore Disk User's
called Texted.
Texted is a simple
which way well fulfil
essing needs.

Texted —
word processing made simple



021
2 FOR PAIRS
4 FOR A FLUSH
TOTAL 6

Cribbage Master

F.S.S.L

FINANCIAL·SYSTEMS·SOFTWARE·LIMITED

BASIC 8

Unleash the hidden graphics power of your 128. At last, your Commodore 128 can rival the 16 bit micros! Imagine your 128 in 80 columns producing a resolution of 640 x 200 in mono and 640 x 192 in 16 colours without any additional hardware! (640 x 400 version available soon.) Sounds impossible? Not with Basic 8, the new graphics language extension.

Basic 8 adds over 50 Hi-Res graphic commands to C128 Basic. Just select one of the many graphics modes and draw 3-D lines, boxes, circles and a multitude of solid shapes with a single command. We've even added commands for windows, fonts, patterns and brushes.

To demonstrate the power and versatility of this graphics language, we have created Basic Paint, a flexible icon-based drawing application program. Basic Calc, a full featured spreadsheet and Basic Write a simple to use word processor with a multitude of fonts Also included is an icon based desk-top utility which provides quick and convenient access to each of your own Basic 8 creations

All this graphics potential is yours for **£34.95** 64k Video RAM upgrade kit **£29.95** and The Beginners Guide to Basic 8 with disk **£19.95** Basic 8 Tricks & Tips Volume 1 **£4.65**

1541 '71 Disk Drive Alignment Package

Problems with your disk drive? The alignment package reports the alignment condition of the disk drive as you perform adjustments. On screen help is available while the program is running Includes features for seed and head stop adjustment. Allows you to test each full and half track as you realign the drive Complete instruction manual on aligning both 1571 and 1541 drives. Even includes instructions on how to load the alignment program when nothing else will load. Works on the C64, SX64 and 128 in 64 or 128 mode **Only £24.95.**

1571 FIX ROM

Are you experiencing any of these problems with your 1571 disk drive Errors when using Superbase? Device Not Present Errors? Major problems when 2 or more open files? Takes forever to recognize a floppy disk? If you have one or more of these problems then you need 1571 FIX ROM, developed by Commodore Inc USA just plugs in, solves over 19 main problems. **Only £24.95.**

Commodore 1581 Disk Drive

The New Commodore 1581 drive is now available for the C128 or C64. This drive offers one megabyte of memory, THREE THOUSAND ONE HUNDRED AND SIXTY blocks free (800K formatted capacity), and a very fast EIGHT THOUSAND Characters per second loading rate Software for this new drive includes, VizaWrite Classic, HackPack, Petspeed, Oxford Pascal, CAD PAK, Chart Pack, Fontmaster, Spellmaster and much more Using a package called Super 81 Utilities most software can simply be copied across to the new format. The drive is available for **£164.95.**

Warp Speed

An incredible accelerator cartridge for the Commodore 128 and 64 Warpspeed has a switch that allows you to select 128 or 64 mode. The features of this cartridge apply to 1541, 1570, 1571 and 1581 disk drives in 64 or 128 mode These features include loading, saving, formatting and verifying 10 times faster, extended DOS commands, print screen, disk and file copy, built in machine monitor, built in track and sector editor and much more **Only £34.95.**

The Big Blue Reader

New from SOGWAP Software Inc., The Big Blue Reader is ideal for those who use IBM PC compatible computers at work and have the Commodore 128 or 64 at home The Big Blue Reader is not an MS-DOS emulator, but rather a unique and easy way to transfer word processing, text and ASCII files between two totally different formats Commodore and MS-DOS. The Big Blue Reader requires a 1571 on both the Commodore 128 and 64 and will not work with a 1541 or similar drive. **Only £34.95.**

The Drive Box

The drive box allows you to easily change the device number (8, 9 10 or 11) of your 1541, 1571, 1570 or 128D computer It will, as a bonus, allow you to bypass the write protect sensor allowing you to write to the back side of the disk without cutting a notch. Installation requires a little soldering **Only £19.95.**

The Programmers Notebook

The Programmers Notebook is a high speed data storage and retrieval system designed specially for the programmer With Programmers Notebook you can develop a special programming information database tailored to your own needs. It will provide a means of indexing your short routines and other programming information Each entry has a brief description, date and type of routine Once entered you can conduct tailored searches on the entire database, in fractions of seconds, looking for any information you select. It will sort, compile, display and provide printed copies in almost any format you desire **Only £24.95.**

Home Designer 128

Forget the name this is not a kitchen planner it is probably the most advanced CAD system for the C128 computer. Home Designer allows you to create extremely detailed and accurate drawings at virtually any scale, with any resolution approaching a plotter It offers accurate scaling, a powerful command set and up to five drawing layers. Why is Home Designer better than the rest? It is the only CAD system which allows layers of diagrams to be drawn and then laid upon each other in any order. Home Designer has an advanced zooming feature, allowing you to actually increase the drawing resolution, unlike other programs which simply display fat dots on the screen. Most dot matrix printers and plotters are supported and Home Designer works with a Commodore 1351 mouse **Only £34.95.**

Super Pack 2

Super Pack 2 is the combination of three packages, Pocket Writer, Filer and Planner The integrated word processor, spreadsheet and database work on the Commodore 64 or 128 in 40 or 80 columns. Each package has all the features you would expect, plus a few you would not. The word processor displays on screen bold, underline, italics, sub and super script fonts as you type The spreadsheet includes a sideways printing feature as well as 16 digit accuracy, plus a graphics module to produce bar, line and pie charts quickly and easily from your data. The database includes a report writer to create your own professional reports. Full details available on request. Super Pack 2 by Digital Solutions Inc is available for only **£64.95** or **£39.95** each.

Super Disk Librarian

Super Disk Librarian will catalogue up to 1,000 disks and 15,200 programs. Do you have a lot of disks full of programs? Have you ever lost a program which you knew you had somewhere? Then Super Disk Librarian will provide you with a simple means of sorting and indexing the file names Once the disks have been catalogued, numerous reports and lists can be printed and the library sorted and searched. Super Disk Librarian for the Commodore 128 **Only £29.95.**

The CP/M Kit & Users Guide

The CP/M Kit introduces and explains the unknown, third mode of the C128. The CP/M Users Guide is a 300 page book by Abacus Software covering all aspects of CP/M. Subjects which include the system disk, resident commands and disk copying are described in detail. The CP/M kit contains over 20 CP/M programs including a word processor, chess game and a disk cataloguing program accompanied by a detailed guide to running programs in CP/M The CP/M kit and Users Guide **Only £34.95.**

Super Disk Utilities

The ultimate utilities disk for the 1571 disk drive and C128 computer Copy whole disks with 1 or 2 drives, change disk format, perform CBM DOS, CP M and MS DOS utility functions, scratch, unscratch, lock and unlock files write protect disk, contains disk editor, drive monitor, unique RAM writer and more **Only £24.95.**

● Commodore 1351 Mouse	£29.95	Commodore 128D	£374 95
Merlin 128 Assembler	£48.95	1571 Disk Drive	£184 95
Chartpack 128	£29.95	1901 Monitor	£210.95
CADPAK 128	£29.95	Anatomy of 128	£9.95
Super C	£39.95	Anatomy of 1571	£9.95
Cobol 128	£29.95	Peeks and Pokes	£9.95
HackPack 128	£14.95	CAD on 128	£9.95
RAMDOS	£29.95	Tricks and Tips	£9.95
Petspeed 128	£14.95		
Word Writer 128	£39.95		
Swiftcalc 128	£39.95		
Data Manager 128	£39.95		
Fontmaster 128	£49.95		

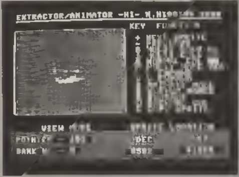
● Order with Access, VISA, cheque or Postal Order, COD and Express delivery available foreign orders add £3.50 per package Prices include VAT and postage Please allow 14 days for delivery

18 High Street·Pershore·Worcs·WR10 1BG·Tel: (0386) 553153

CONTENTS

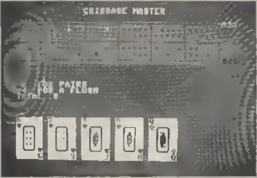
CONTENTS
Commodore Disk User
Volume 2 Number 1
November/December 1988

IN THE MAGAZINE



Extractor

ON THE DISK



Cribbage Master

Editor: Stuart Cooke
Deputy Editor: Fin Fahey
Confectionery Procurement and Artwork: Alan Batchelor
Advertisement Manager: Marcus Collingbourne
Copy Control: Andrew Selwood
Orignation: Ebony Typesetting
Distribution: S.M. Distribution
Printed by: Chase Web, Plymouth

ARGUS
PRESS
GROUP

Commodore Disk User is a bi-monthly magazine published on the 3rd Friday of every alternate month. Argus Specialist Publications Limited Commodore Disk User 1 Golden Square London W1R 3AB Telephone 01-437 0626 Telex 8811896

Opinions expressed in reviews are the opinions of the reviewers and not necessarily those of the magazine. While every effort is made to thoroughly check programs published we cannot be held responsible for any errors that do occur.

Update

More dispatches from the front line

Quality not quantity

Digital Integration and Novagen are two companies bound by a common motto of quality not quantity. This Christmas they each have products which have taken over a year to complete.

If depth of research equated with the period of weeks in the charts, then DI's F16 Combat Pilot would stay there for several years to come. The company are bold enough to claim that this program is as near to flying the real thing as many of us will ever get.

It's not surprising that they make such a claim when you consider that the research has been thorough enough for them to seek advice from the RAF, USAF and one of Britain's foremost authors on modern aerial warfare, in addition to playing every F16 flight sim in sight.

Novagen's offering is Battle Island which has the distinction of being the first game from the company that has not been authored by Paul Woakes. Well, that's not entirely true because Paul did write the core routine which allowed programmer Gary Walton to code in over 350 detailed screens with multi-directional scrolling!

Battle Island – Novagen

Not a Konix

Joysticks vying for the limelight with Konix include the Micro Blaster from Replay (marketed by Compumart) and RP Products' Mister Joystick Crystal Range

Compumart's stick claims to offer the widest range of features for any joystick costing £12.95. So the blurb goes, "ergonomically designed in red and black with non-slip rubber feet for easy use. The eight higher quality micro switches..." and so on until it mentions the 'rapid fire button', '1.4 metre cable', 'steel shaft', et cetera, et cetera.

The RP Products info shows similar lack of imagination I quote, "the Crystal Joysticks feature colourful internal workings housed inside a glass-clear case with a bright red pistol grip handle fitted as standard", "five year guarantee", "priced at between £15 and £20", drone, drone

Come on lads and lasses, how about a bit of imagination. Remember the photo of some berks from as far apart as Belgium and Brighton, one winning a prize for guessing the results of the 'waggle' test, coyly posing in front of a statue of a widdling boy. Give us some laughs. We know what a joystick is (and the ones in quesbon are undoubtedly very worthy products), now let's make it interesting, just like Steve Davis .isn't.

Compumart's rigorous whatsit

Sega-Genic

Mediagenic, the born again Activision, has signed up conversion rights to five Sega titles. Galaxy Force, Altered Beast, Hot Rod, Sonic Boom and Ace Attacker are the five titles in question and launch labels will be shared between Activision and Electric Dreams.

European Vice President of Mediagenic, Rod (could that be Hot Rod) Cousens is understandably excited and sees 1989 as being a "blockbusting year for Mediagenic".

Fortran for 64?

Those readers who are interested in the FORTH program on this month's disk may also be pleased to learn that Abacus has released a Fortran emulator in the States

Although Fortran has been around for 30 years, it is still widely used for scientific and business purposes. Abacus' Fortran 64 includes a built-in editor, compiler and linker to enable fast running programs to be compiled and then run without the resident Fortran program.

This product lines up alongside other Abacus language emulators such as Cobol, C and Pascal. It costs $39 95 but there are no plans, as yet, for its release over here in the U K

Lancelot rides out

Magik Knights

Mandarin Software is preparing to launch its second title which follows slack on the heels of the company's release of some time ago, the Time & Magik trilogy.

Lancelot follows the adventures of the ill-starred knight and adheres closely to the plot of *Morte D'Arthur* by Sir Thomas Mallory (another unfortunate knight).

The fact that the adventure is penned by Pete Austin of Level 9 augurs well for its accuracy and quality because he is something of an authority on the Arthurian legends.

Electronic Mart

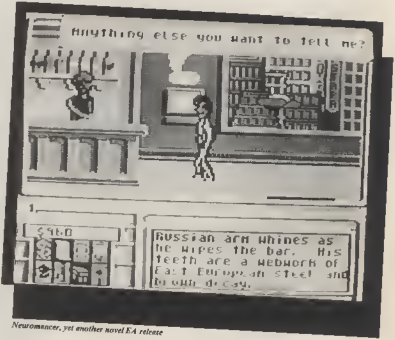
Electronic Arts seems to be bombarding the Christmas market with so many new products that, were it some other company, I'd be tempted to say that if you throw enough of it about, some sticks. However, these champions of the country's disk users seem to have an excellent range of goodies.

Bard's Tale freaks...this office is full of them. "Have you tackled the Medusa yet?" "Oh no, my armour class is too low and besides mummy doesn't like me getting stoned with strange girls" . As I was saying Bard's Tale fans will be pleased to hear of two new role-playing games Deathlord and The Mars Saga

The aim of the first game is to find a way to ooot the Deathlord's hordes out of the land of Lorn (how about a sign saying 'Keep Off The Grass'?) The game map is bigger than anything in any of the Bard's Tale trilogy covering several continents and 157 dungeons Players can transfer their favourite characters from a Bard game, Wizardry or Ultima III

In the Mars Saga the player adopts the role of a bounty hunter trying to earn enough dosh to escape the wretched place.

Amongst the other releases is The Commissioner's Disk which offers everything a baseball manager could want. It's nice to see EA looking after minority interests in this way and our almost civilised Celt, Fin Fahey, is hoping to see a hurling simulation soon. Don't hold your breath, Fin.



Anything else you want to tell me?

Russian arm whines as he wipes the bar. His teeth are a webwork of East European steel and brown decay.

Neuromancer, yet another novel EA release

DISK INSTRUCTIONS

Before you use your disk for the first time read this

We have done our best to make sure that Commodore Disk User will be compatible with all versions of the C64 and C128 computers and their associated disk drives.

Getting the programs up and running should not present you with any difficulties at all, simply put your disk in the drive and enter the following command

```
LOAD "MENU",8,1
```

Once the disk menu has loaded you will be able to start any of the programs simply by pressing the letter that is to the left of the program that you want to use

C128 users please note that you should be in C64 mode when using the disk You can enter C64 mode by either

i) Holding down the Commodore key (bottom left of the keyboard) when turning the computer on or,

ii) After turning the computer on type G064 and answer "Y" when prompted "ARE YOU SURE?"

It is possible for some programs to alter the computer's memory so that you will not be able to LOAD programs from the menu correctly until you reset the machine We therefore suggest that you turn your computer off and then on before loading each program.

How to copy CDU files

You are welcome to make as many of your own copies of Commodore Disk User programs as you want, as long as you do not pass them on to other people or worse, even sell them for a profit.

For people who want to make legitimate copies, we have provided a simple machine-code file copier To use it, simply select the item FILE COPIER from the main menu The copier works with a single drive, is controlled by means of the function keys as follows

F1. Copy file - the program will prompt you for a filename

F3 Resave the memory buffer - you may get an error on a save (perhaps you left the drive door open) Use this to try again

F5· Disk commands - allows you to enter any regular C64 disk command

F7· Displays the directory

F2·Exits the program and returns you to Basic

Disk Failure

If for any reason the disk with your copy of Disk User will not work on your system then please carefully re-read the operating instructions in the magazine

If you still experience problems then

1) If you are a subscriber, return it to
INFONET LTD
5 River Park Estate
Berkhampsted
Herts. HP4 1HL

2) If you bought it from a newsagents, return it to·
Disk User Replacements (BBC or Commodore as appropriate)
DISCOPY LABS
20 Osyth Close
Brackmills
Northampton NN4 0DY
Telephone 0604 760261

Within eight weeks of publication date disks are replaced free

After eight weeks a replacement disk can be supplied from DiscCopy Labs for a service charge of £1 00 Return the faulty disk with a cheque or Postal Order made out to DiscCopy Labs for £1 00 and clearly state the issue of Disk User that you require. No documentation will be provided

Please use appropriate packaging, cardboard stiffener at least, when returning a disk Do not send back your copy of the magazine only the disk please.

Don't miss...

CDU's Bumper New Year Issue

Think we've packed a lot on the disk this issue? Just wait till you see January's CDU This time, the disk will have not one, but two sides packed with value-for-money software. Side One will feature our usual wide range of programs, from business programs to computer languages, while Side Two will be filled with games of all sorts. This is the issue you can't afford to miss, so get on to your newsagent, or better still, get a subscription now!

Apologies

Due to a mistake when crunching the Addit program on last issue's disk, this is missing several lines, and will not run. A new version has been provided on this issue's disk.

The all new range of Commodore Diskettes

Don't risk losing your valuable data. The superb quality offered by Commodore Diskettes ensures perfect data storage, time after time at a price you can afford.

Available now from your local computer dealer

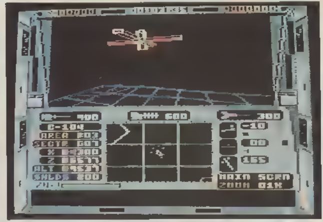
Commodore

The Original Diskette - Higher Quality Lower Price

Trade enquiries only to SJB Disks Ltd., the officially appointed sole U.K. distributor

Reviews

Our team of intrepid reviewes analyses the latest batch of computer entertainments



Echelon

Echelon is not only a 3D space flight simulator that offers a massive vector graphics planet to explore but also introduces a new control method Forget the keyboard, mouse or joystick the Lipstick is here

The Lipstick is a headset unit that plugs into a joystick port and consists of a pair of headpnones that don't do anything except position the Lipstick or microphone. This gives the game a voice activation system through which you can launch missiles and plasma bolts by simply yelling FIRE! In fact you can wipe out targets by saying anything but launching a missile with the command GONK distracts from the gameplay

The Lipstick is only part of the game control system as you also use a joystick to fly our craft and most of the keyboard to activate its systems, so it's just as well a Gunship style keyboard overlay is crammed into the gamebox along with the Lipstick, gamebox and 72 page instruction manual.

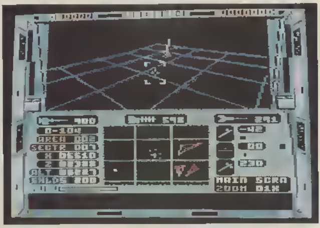
The craft you fly is a Lockheed C-104 Tomahawk and is described as the most awesome combat and exploration vehicle to operate in the 21st Century Luckily, it's as easy to fly as 20th Century helicopters and planes and has standard thrust, pitch and bank controls and so it won't take long to be proficient enough to take on one of the game missions

The main game is a space adventure exploration style game in which you must discover what has been happening on this planet by searching out, teleporting aboard and examining curious artifacts that you will find throughout the massive game area. Here you'll find the Echelon training obstacle courses and target ranges and further afield ancient dams, bridges, mines and cities, a giant radio telescope and remains of the McAdams Penitenary

As you explore these areas you may begin to piece together the mystery behind this patrol area and even decipher the curious pirate maps

If you find all this too much to handle you can always contact the datalink computer, change one of the parameters and slog it out with wave after wave of alien ships which exercise your Lipstick and the buttons to select either missiles, plasma bolts or laser

Echelon is a massive game of epic proportions and you will need to like the longer more adventurous style of game to get the most out of it. The combat game can only be described as some light relief as you can fight only one type of alien craft and so will rapidly become tiresome. TH



At a glance.

Title: Echelon.
Supplier: US Gold, Units 2/3 Holford Way, Holford, Birmingham, B6 7AX
TEL: 021 356 3388.
Price: £14 99
Graphics: 3D vector graphics.
Sound: Thrusters and explosions.
Playability: Takes some time to get the most out of the game.
Addictiveness: The space simulator version of an adventure.

Barbarian

It was a shock when Hegor returned home to find his father fighting for his life against a huge dragon. It was a hopeless cause but at least his father managed to lure the dragon away from the village before he succumbed to the fiery breath. Now an orphan, Hegor vowed revenge and left the village in search of adventure.

Over the years, Hegor developed his prowess as a Barbarian and his feats of dragon slaying, drinking and womanising became legendary throughout the land. Rumours eventually started to filter through to him of a new evil spreading across the country, an evil that seemed to be particularly virulent near his home village. The name Necron was whispered and eventually the King offered his kingdom to anyone who could destroy this evil once and for all.

Hegor returns home and discovers an old man who turns out to be the ghost of his father. He tells Hegor a convoluted story about how Necron is in fact his twin brother whom he should have killed when he had the opportunity years ago. Instead though, he spared him and now look what has happened. So it is that our intrepid hero sets off on his quest.

Hegor is controlled via a series of icons at the bottom of the screen. To add further to your fun, the instructions cunningly don't bother to tell you what they are – unless, that is, you happen to speak French, German or Italian. Even then, the French instructions have bits missing. For any non-linguists amongst you, you have the option of making Hegor move left, right, up and down, stand still, jump, run, attack, defend, run away (surely an option that no real Barbarian would ever use), pick up, drop and use an object.

The game itself is presented as a series of screens, most of which contain one or more hazards that have to be overcome before access to the next is allowed. Typical problems at the start of the game include fighting an assortment of monsters, avoiding lumps of falling masonry, leaping over collapsing bridges and dodging sniper archers that appear from under trap doors. All fairly routine Barbarian stuff. As the game progresses, you have to discover a means of killing the same dragon that saw off your father all those years ago until you come to the final conflict with Necron himself. Defeating him causes a volcano to erupt and so you have only a limited amount of time to make good your escape and claim your prize.

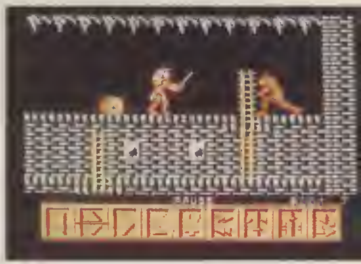
When this game first appeared on the Amiga, written by Psygnosis, the workload in the office went right down as everybody wanted to play. The game looked great and had a definite 'just one more go' addictiveness to it. The conversion to eight bit formats has been done by Mastertronic on their Melbourne House label and I am sorry to say, it is rubbish.

On the original, there were some stunning loading screens to enjoy. On the 64 version, you get to play Space Invaders. I suggest that you look at the shot on the packaging to give you some idea of what you have been missing. The game itself plays like a wet blanket. The monsters move with all the menace of My Little Pony, the controls are sluggish to the extent that when you flick from one screen to the next, you have to wait ages for the joystick to catch up and point to the icon you want.

Even given the different capabilities of the eight and sixteen bit version, the graphics and sound effects on the C64 are dire. The graphics especially are fuzzy in the extreme.

This conversion proved to be a real disappointment on a game I was looking forward to playing. As a hero, this Barbarian couldn't knock the skin off a rice pudding. He certainly couldn't fight his way out of the box and I suggest you add a few extra elastic bands to prevent his accidentally escaping.

GH



At a glance.

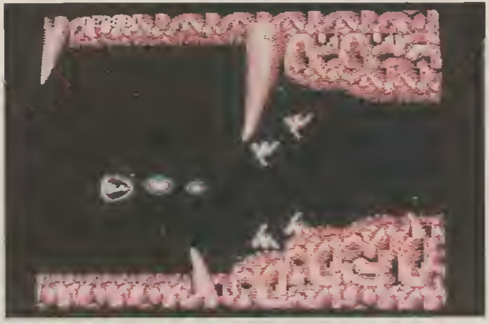
Title: Barbarian
Supplier: Mastertronic/Melbourne House
Price: £14.99
Graphics: Its like looking at the screen with wool over your eyes
Sound: As above but with wool in your ears
Playability: Definitely woolly
Value: If you buy this, someone has definitely pulled the wool over your eyes

▶ ▶ ▶ ▶ ▶ ▶ ▶

Salamander

When Nemesis was released a couple of years ago it was hailed as the best coin-op conversion Salamander was the arcade best selling sequel which is now available for the C64 and it's even better.

The Salamander is an evil dictator that rules the galaxy but now you and two other colleagues step forward to challenge its might.



Piloting a small and fragile single laser ship you enter its maze of terror. Ahead of you lie the Caverns of Desperation packed with traps and monster and at then end of the fourth frantic level a final conflict with the Salamander's brain

Heroes are made of stern stuff and so you won't mind being desperately outnumbered and the fact that many have gone before you and all have failed will just add to your challenge What about the Nucloed Spider that revolves hurling a barrage of missiles at you from each of its tentacles or the claws that reach out from the rock to grab you? Well, it's too late to change your mind as you've already volunteered

As you enter the first cavern you are met by a reception committee in the shape of waves of alien fighters – these are usually easily despatched or avoided before entering the level itself Now you have to stay sharp and things will start happening quickly. Suddenly claws start appearing which are deadly to the touch, as are the cavern walls and ceiling which many a pilot will discover while avoiding the claws Then when you think you've got the hang of those organic monsters the rock begins growing to narrow the cavern passageways and crush you Only swift reactions and a steady hand get you out of trouble only to find more alien waves, giant gas bubbles that explode when you come in range and solid rock that must be blasted to find your way through (before it grows back).

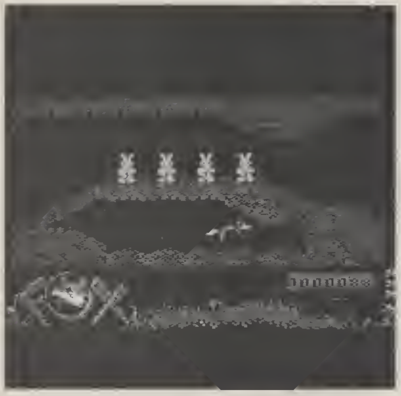
Luckily, you do have some help in the form of bonus pods that you can collect to add more frequent laser fire, missiles that hug the cavern's room or floor and destroy anything lurking there and up to three multiples that follow your every move and and multiply your firepower.

At the end of each level is a giant monster that pursues you around a single screen. There's no avoiding this one, until either your game ends here or you manage to find its weak spot and pile in enough shots on target to destroy it.

One of the best coin-op conversions. **TH**

At a glance ■ ■ ■ ■

Title: Salamander
Supplier: Imagine, 6 Central Street, Manchester, M2 5NS
TEL: 062 832 6633.
Price: £9 95.
Graphics: Superb animation but can get quite slow at times
Sound: A tune to blast by and a few explosions
Playability: Easy to learn but impossible to master.
Addictiveness: I've got to have another go.



Foxx fights back

A refreshing idea for a game, this I am completely in agreement with Oscar Wilde ('the unspeakable in persuit of the uneatable') when it comes to the subject of fox-hunting Considering that the English are supposed to favour the underdog, the spectacle of 200 stockbrokers and estate agents pretending to be country squires trying to kill one poor furry animal is contradictory to say the least.

As the title says, Imageworks' Foxx, however, fights back This fox has firepower. Anything from an automatic pistol to a machine gun is available to blow away those hunters. Of course, there are the Hell's Beagles to look out for. These have mobility on their side, and once they're on their Harleys, they'll give you a run for your money.

Main motive for Foxx to run around the countryside is the same as for any wild animal – the eternal search for grub Foxx's Vixen is waiting in the Earth, and she's not happy to see you come back empty-handed. Food can be found, at some risk to life and limb, in the form of sausages, apples, pies and so forth in various locations that you will need to jump to reach.

Besides the collectable food that you can take back, you will need to find edible substances to keep your own energy going for the search. This food is on the hoof, and takes the form of coops full of nesting chickens or burrows full of cute little lovable bunnies. Yum, yum!

Stamina left is shown by a little fox icon at the bottom left of the screen. The state of Foxx is shown by the length of his tongue on this 'pantometer' The longer it is, the closer to death you are.

The game is basically a simple running, jumping, shooting kind of thing. I normally like to play something a little more complex, but I found this game quite captivating It's witty, the action is smooth and convincing, and the scrolling backgrounds are varied and watchable Great fun, if a bit lightweight. **FF**

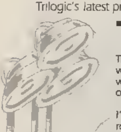
At a glance ■ ■ ■ ■

Title: Foxx fights back
Supplier Image Works, Headway House, 66-75 Shoe Lane, Lohoon EC4P 4AB **Tel:** 01-377 4645 **Price:** £12 99
Graphics: Nice backgrounds, - a good little mover
Sound: Hunting toons
Playability: Getting Foxx to jump a little tricky, but generally smooth
Addictiveness: Pass me another bunny rabbit. Crunch!

REVIEW

Drive Doctor

Disk Drives have finely tuned mechanisms so is Trilogic's latest product a boon or a bane?



by Eric Doyle

The Drive Doctor is the second repair kit from Trilogic, following hard on the heels of their comprehensive Datasette Doctor The new utility consists of a disk and cassette with a very small manual which made me wonder whether more bad than good would come of this package.

Unlike every other disk alignment kit that I've seen, the Drive Doctor's diagnostic routines are supplied on cassette. Let's face it, if your drive's up the creek a disk based program is not going to help a great deal! The disk included in the kit is merely an accurately aligned test disk for use with the cassette program and doesn't contain any programs itself.

The program has five tests which check the drive's speed, back stop setting and alignment, as well as ensuring that the head movement is not jamming and that the stepper motor hysteresis is accurate

The instruction sheet explains in five simple steps how the drive can be stripped down ready for testing and it's at this point I issue my first warning Unlike the computer, the disk drive mains transformer is built into the chassis This means that there are a few wires with 240V skulling around them. Although most of the drives that I've seen are well insulated some may not be, so always check that the mains connectors are not worn before plugging in. Remember that the chassis is metal and you may be the most appropriate earth lead that the killer volts can find. I would have preferred it if Trilogic had added another page to cover electrical safety procedures to protect their customers.

My second criticism is that, in my experience, head alignment is often the least likely cause of loading errors. I would rather that Trilogic had organised the booklet in a routine which ended with alignment checks rather than including them as the second procedure in the manual. As a guide to potential buyers I will consider the features in what I consider to be a logical, diagnostic order.

Most faults can be attributed to three root causes: disk speed, head jam and back stop displacement. Each fault has its own diagnostic test and I'm convinced of Trilogic's claim that the test is almost as accurate as the standard oscilloscope tests.

The disk speed is altered by a small screw on the underside of the PCB mounted near the front of the chassis Simply by inserting a small screwdriver into the only apparent slot in sight, the speed can be varied and the screen display shows the change with a figure from 0 to 9. When the zero constantly appears on the screen the speed is set. This may be all that is required to get the drive back into working order but if it is not, the second test can be tried.

The head jam test simply causes the head transport to scan back and forth across the face of the disk in a continuous motion. If there is any problem with the transport mechanism it can be easily observed as a juddering motion The solution is lubrication.

On reaching the third level test, the drive mechanism has to be carefully eased forward in its mount to allow access to the back stop. This is the sliver of metal that causes the unnerving knocking sound that first emanates from the drive when a disk formatting command is executed. Although the adjustable stop is well and truly laquered in position, the constant battering may eventually alter the position. This is crucial because the disk drive uses this position as a reference point for finding all of the drive sectors – if it moves a mere millimetre it will cause problems.

The horrendous noise created by the back stop leads some aesthetes to fit a 'soft' stop which is a wire assembly that absorbs some of the impact of the blows. Think about the logic of this. The drive hits the stop to find the correct registration of the tracks but the soft stop bends to absorb the blow. As Trilogic point out, this is a recipe for disaster – soft stops are out unless you enjoy recalibrating your drive at regular intervals

Maybe the motor is defective? It can be tested by the hysteresis test which merely checks to see that the motor returns the head accurately to a given track after hitting the back stop. If the head jam test showed no problem but the hysteresis test fails, it's a workshop job I'm afraid.

Finally we come to the alignment test. This may mean a lot of fiddling about to get the stop and the motor in the correct relative positions and is not a test for the faint-hearted

Using the Drive Doctor I managed to get two of my drives back into working order. A third drive failed to respond properly and the manual suggests that the fault must lie in the circuitry – they were right So the financial calculation is £30 per repair which for three drives would have amounted to £90. Since two were repaired with the Drive Doctor I saved £60 which is far more than the £14 99 I paid for the Drive Doctor. So I'm already showing a handsome profit. Thank you, Trilogic, for the best disk drive alignment kit that I've seen Sort out your manual and you've got a world-beater.

The Drive Doctor retails for £14.99 from Trilogic, Unit 1, 253 New Works Road, Bradford BD12 0QP.

Warp Speed

A new contender in the cartridge wars, Warp Speed is the only one to offer dual C64/128 modes

By Gordon Davis

Warp Speed has the distinction of being the first cartridge that offers both C128 and C64 compatibility It's going to have to compete with a fair coterie of contenders CDU recently reviewed (July/August) all the current competitors in the area, from the Expert to the amazing Warp 25.

This latter received the speed accolade in our comparative test. I'd like to have put Warp Speed through an identical procedure, but I didn't have the precise test programs available

I did however run some tests on a 47-block Basic program of my own With no assistance from the cartridge, this program loaded on a 1541 drive in 35 seconds. The manual claims that Warp Speed can load such a file roughly 500 per cent faster. In fact, it loaded the file in 10 6 seconds, just over 300 per cent faster.

You can improve on this, however The fast load on Warp Speed writes files to disk in 1571 format on a 1541 drive Using the cartridge these can be reloaded, the manual says, up to 1000 per cent faster.

The fast save itself takes just over 7 seconds, while reloading takes 6 seconds in this format. That's not 1000 per cent, more like 600, but to be fair, the claim is *up to* 1000 per cent faster.

This, frankly, is rather sluggish. Warp Speed is doing the bare minimum of fast loading. You see, most rival cartridges use more or less sophisticated crunch/decrunch systems to reduce the number of bytes that they have to read and write. As a result, the Action Replay using Warp 25 is able to do things at an incredible 35.6 bits per second That's approximately 2000 per cent faster than normal about three times as fast as Warp speed.

Warp Speed at its fastest is loading at about 10 bps (as against 1.8 bps for a naked system) which makes it comparable with the Freeze Machine in its slowest mode Since this latter was the slowest load in cartridge speed trials, it's not a good performance.

Clean cut

But perhaps Warp Speed has other good points? Well if you're on the lookout for a cartridge that does naughty things don't look at this one You can't grab anything, freeze it or twiddle with it in any way. Cinemaware have kept this little box squeaky clean, so unlike some other cartridge companies, they won't be getting those nasty solicitor's letters

Of course, this makes Warp Speed a little worthy-but-dull. Nonetheless it does have useful features Disk and file copy and multi-scratch commands are included, plus a very respectable Sector Editor, and an extensive range of utility commands

There's also a built-in machine-code monitor Cinemaware says 'one of the most advanced monitors ever produced for the Commodore 64 and 128' Could've fooled me. I must be missing some little thing or other, because to me it looks just like a monitor, neither better nor worse than a host of other monitors. Worse, the manual describes it as a monitor/assembler An incredible achievement, because what we have here is an assembler with one command! In fact what it does is take a mnemonic and operand in immediate mode and convert them to machine code, that's all Useless for entering more than about 8 bytes of code, and not even that if you need to branch. Hype, us?

All things considered, the most useful thing about Warp Speed is that it has a little switch which enables it to work in either C128 or C64 mode Apart from that it's a rather dull, pedestrian product far surpassed by other cartridges on the market.

CDU Forth

Bored with Basic? Try this powerful high-level language

By R. Lincoln

Alternatives to Basic on home micros of the C64's generation have always been eagerly sought, considering Basic's limitations and structurelessness. One such alternative is Forth, one of the few languages to share with Basic the distinction of being incorporated into a home computer rom, as the basis of the ill-fated Jupiter Ace, a machine that may have deserved better from the market.

Our version will give you a taste of this powerful language, although you may find it unaccomodating to the beginner, largely due to its use of the much-feared Reverse Polish Logic. We have no space here to give a full description of the language, so we suggest that budding Forth users should seek guidance at their nearest computer bookstall

You will, however, need a command list, as this FORTH implementation has been adapted for the C64, and contains some non-standard commands for file handling and other functions. All C64 serial I/O has been implemented.

In addition, an extension has been added to this FORTH enabling it to handle standard C64 floating point numbers. Normal FigFORTH is an integer language, using 1, 2 and 4 bit integers. To adapt the package, an extra floating stack has been added, and commands have been provided to convert between integer and floating point

To get started outside the CDU menu, type LOAD "FORTH",8,1. The system will then autorun. You will need to allocate some text buffers when the title screen appears. The command for this would be, for example, '5 BUFFERS'

Having allocated the buffers, you will then need to enter 1 LOAD to load in FORTH extensions from disk plus the screen editor

Once the system itself is loaded if you want to do any editing the editor must be loaded from the disk. Before doing this I will explain how programs are stored. FORTH uses a system of virtual memory which means the programs are stored as screens which are held on disk until you want to see them, one disk can hold about 160 screens of information When a screen is required either to edit or read it is read into buffer in the computer and when finished with, if it has been updated it is read back to the disk. You can allocate as many buffers as you need up to a maximum of about 40 (because you run out of free memory), the more buffers you have the less space there is for programs, ten is about right. The buffers are allocated by typing 'n BUFFERS' where n is the number of buffers you require, then a screen is loaded by typing 'n load' where n is the number of the screen you wish to load. This screen will then be retrieved from the disk and interpreted as if you had just typed in what was on it.

To load the screen editor and extensions type '1 LOAD' after allocating some buffers, this will load in some new words and the screen editor from screens 1 and 2. (Don't practice editing on these two or you will find you've got no editor anymore!)

The editor which is written in FORTH (as is the Interpreter and the Compiler) is very simple as it was about the first FORTH program I wrote.

The FORTH editor

The screen editor is written in FORTH itself, on screen 2, and can itself be edited and added to.

Editor commands

EDITOR (to enter the editor vocabulary)
n LIST (to list screen no to edit)
EDIT (to enter the screen editor. If the screen is being used for the first time it will contain garbage which is cleared by typing 'WIPE')
L (lists the current screen)
JUST L<, or > (lists the previous or next screen respectively)
O n (deletes line n from current screen)
I n (inserts a new line n, the rest of the screen is scrolled down and the last line is lost)
n text (places text on line n of the current screen)
UPDATE (if UPDATE is not typed the screen will not be saved, if it is then the screen will only be saved when the buffer it is in is required by the system. If you turn off the machine at this point the screen will not have been saved)
FLUSH (FLUSH sends ALL updated screens back to the disk overwriting any data held there. This is usually used just before power down to save all data)
FORTH (to return to the FORTH vocabulary when finished)

WIPE Clears and sets up current editing screen

Editing mode

When in screen editing mode, a '.' will appear on the left of each line to remind you that this mode is activated. Lines are entered by first typing in the line number followed by a space and up to 38 characters. An example would be:

.01 TEST ON UNE 1

If an 'I' is entered before the line number, the line will be inserted and the rest of the screen moved down to accomodate it. Entering 'E', followed by the line number will display the line on the screen ready to be edited.

Apart from these commands, editing a FORTH line is exactly the same as editing a similar line in Basic

Input methods and storage

When the system is first loaded you will be confronted with a title screen an 'OK' and a cursor flashing below it. FORTH is waiting for input of some kind, it could be in two forms

1) A numeric value, FORTH mostly uses integers so that processing speed can be kept high, these must be in the range -32768 to 32767 (displayed using .) or 0 to 65535 (displayed using U.). If these ranges are not large enough FORTH will also recognise double precision numbers which take up twice as much room as normal ones. Double precision numbers are entered including a full stop somewhere in the numger e.g. 56.7. This is stored as 567 (the point only signifies a double number NOT a decimal). You will find double numbers take up two positions on the stack and can store values up to about 2000 million.

2) A Word, this can be any sequence of characters that FORTH can find in its dictionary e.g. VLIST, DEPTH,.,.!, BUFFERS etc (The dictionary can be listed to the screen by typing VLIST. Very often FORTH words are no more than single characters but they are still words).

Programming

To program in FORTH you just define new words for the dictionary to do the purpose you want them to do. If I wanted to print my name on the screen I could type In'.(Richard Lincoln)' which would work, but my hands would wear out, so I can define a new word to print my name: I will call it NAME but I could have just as easily called it HOUSE, CAT, £, etc. there are no restrictions other than it must be less than 32 characters long. So I want to define NAME so that when I type it in it displays my name ':NAME." Richard Lincoln";' If you type in that and the type VLIST you will see that the word NAME has been added to the dictionary and FORTH will now know what to do if you type it in (it displays the message Richard Lincoln). To define a new word just type a colon followed by a space then the name of the new word then the list of constructions you want it to do followed

Standard FORTH

[illegible]

by a semicolon. The best thing about FORTH is you can now use these new words to define other words and-so-on until you have a program eg:

:NAME. "Richard Lincoln"; (Prints my name)
:NAME 0 DO NAME LOOP; (Prints my name a set number of times eg. 5 NAMES will Print it 5 times)
:NAMES-FOREVER BEGIN NAME AGAIN; (Prints my name for ever, or until you Press runstop-restore)

You can type in the words in brackets but FORTH will ignore them since brackets signify a comment, don't forget spaces either side of all words.

Buffers are allocated above the Dictionary then words are placed above these moving up in memory. If you manage to use up the whole 40 + k then the system will crash as there is no check on memory usage (to keep speed up).

Disk formatting

The screens are stored as a relative file on the disk Each screen takes up four disk blocks. If you want to use your own disk, format it in the usual way either from Basic or from FORTH 'DISK N:NAME,ID'. Then allocate some Ram buffers and type 150 BUFFER this will find a buffer in memory and label it screen 150, then type UPDATE FLUSH, this will make FORTH save it to disk. If the disk is empty a new relative file will be set up with 150 screens in it (this may take some time), you can now load, clear and edit them at leisure.

Reverse polish notation and the stack

Forth uses a number system known as Reverse Polish This may seem a bit weird at first but it is well suited to the use of stacks (on which the whole system is based)

When a number is typed into the computer followed by Return the system will respond with 'OK' on the same line, what has happened is the number has been stored on the top of the parameter stack (usually refered to as just the stack) The stack is where values and parameters are passed to words and programs, it is analogous to a pile of bricks - when you type in a number it is written on a brick and placed on top of the pile, so if you now type in another number and press Return it will be placed on top of the last one so you have a stack of two numbers, the most recently entered being on the top. You can continue entering numbers as long as you like and they will just pile up with the most recent at the top and the oldest at the bottom (there is room for about 120 numbers before the system may decide to forget some of them)

Now you have all these numbers hanging about it might be a good idea to do something with them. If you now type in '.' and Return the last number you typed will be displayed followed by OK on the same line The word '.' takes the top number off the stack (or top brick off the pile) and displays it on the screen. When the system finds itself at the bottom of the stack i.e it has no more numbers stored (no more bricks) it will tell you by saying 'STACK EMPTY?', error messages can allways be recognised because the last character is a question mark If you made a mistake earlier by typing in the wrong word eg 'HELLO' FORTH would have replied with 'HELLO?' because it doesn't know what hello means.

There are words in the Forth vocabulary which enable you to keep track of what is going on in the stack, I will explain a few of them here.

DROP removes the top value on the stack, it throws away the top brick

DEPTH returns the number of values on the stack, so if you typed in one number DEPTH would put one on the top of the stack - which can be displayed using

SWAP takes two numbers off the top of the stack and puts them back in reverse order.

Now we come to Reverse Polish Because the way values are stored on the stack the numbers are always entered before the arithmetic operators eg if you wanted to add 57 to 7 instead of typing PRINT 56 + 7 as in BASIC you would type '56 7 + .'. What happens is the 56 is placed on the stack and then the on top of it so you have two number on the stack at this point, then the plus sign takes the top two values off the stack (57 and 7) adds them together and puts the answer back onto the stack, so the stack now has one value on it (63) this value is the removed and displayed by the '.' Therefore to add 103 to 79 you would type '103 79 + .' and to add 89, 34 and 3 you would enter '89 34 + 3 + .' or '89 34 3 + + .'. The second entry does exactly the same as the first but in a different order, first it puts the three numbers on the stack and then the first plus sign adds the top two and leaves the result on the stack, so that there is now a 37 on top of the 89, then the second plus adds these two together to obtain the result.

The system is well documented in FORTH books and any further help required should be obtained from these

Floating point commands

[illegible]

FORTH commands

This is a very sketchy command glossary, not a language description. Its only function is to serve as a fast reference For a full version, you'll need to get hold of some further reading matter.

Do not read these commands as you would a C64 Basic list. Since FORTH uses RPL, operands need to be placed on the stack before they can be processed. Likewise any results will also be placed on the stack.

The format representing this is, in general: *i1, i2.. * o1,o2..,* where *i1, i2* etc are inputs and *o1, o2* etc outputs. The asterisk represents the command. Some commands, of course, have no operands and/or results.

ABORT Clears the stacks and enters execution state

ABS *i1 * o1* Leaves the absolute value of *i1* as *o1*

AGAIN Used in loop as terminator following a BEGIN command. Used in colon definition.

ALLOT *i1* * Adds the signed number to the dictionary pointer DP.

AND *i1,i2 * o1* Performs a bitwise logical AND on *i1,i2* and leaves the result in *o1.*

BASE This is a user variable which contains the current number base for I/O conversion

BEGIN This is used in a colon definition, and is a loop start. The full loop can take the formats:

BEGIN AGAIN
BEGIN . UNTIL
BEGIN . REPEAT

BL This constant leaves the ASCII code for space.

BLK A user variable containing the block number being interpreted

BLOCK *i1 * o1* Leaves the memory address of the block buffer containing block *i1.*

C! *i1, i2* * Stores the 8 bits of *i1* at address *i2.*

C, *i1* * Stores 8 bits of i1 into next available dictionary byte

C @ *i1 * o1* Leaves the 8-bit contents of address *i1* in *o1*

CHKIN *i1* * All further input is taken from file *i1.*

CHKOUT *i1* * All further output is sent to file *i1*

CLOSE *i1* * Closes file number *i1*

CMOVE *i1, i2, i3* * Moves *i3* number of bytes from adoress *i1* to address *i2*

COLD Cold start

COMPILE Places execution address of the word following COMPILE in the dictionary

CONSTANT Used to define FORTH constants

CONTEXT * *o2* Gives a pointer to the first vocabulary for dictionary searches.

COUNT *i1 * o1,o2* Leaves byte address *o1* and byte count *o2* of message text beginning at address *i1.*

CR Transmits a carriage return to an output device

CREATE Creates a dictionary header for a FORTH definition

D+ *i1,i2 * o1* Leaves double number sum of two ddouble numbers.

D+- *i1,i2 * o1* Applies sign of *i2* to *i1*

DABS *i1 * o1* Leaves absolute value of a double number

DECIMAL Sets BASE for decimal.

DEFINITIONS Sets CURRENT vocabulary to CONTEXT vocabulary.

DLITERAL Compiles stack double number into a literal

DMINUS *i1 * o1* Converts double number to two's complement

DO Loop initiator in form either:

DO. . LOOP or
DO. +LOOP

DOES Defines run-time action within a high-level defining word

DP The dictionary pointer.

DPL Contains the number of digits to the right of the point on double integer input. A user variable.

DROP *i1* * Drops a number from the stack

DUP *i1 * o1, o2* Duplicates number on stack.

ELSE Part of conditional Occurs within colon definition IF . ELSE THEN

EMIT *i1* * Sends ASCII character *i1* to output device

EMPTY-BUFFERS Zeroes the contents of all block-buffers and marks them as empty

ENDIF See THEN

EXECUTE Execute definition

EXPECT Send characters from terminal to address.

F >1 * *o1* Takes top number from floating stack, converts it to a single integer and places it on the stack.

I< F *i1* * Takes single integer, converts it to floating point and puts it on the floating stack

F! Same as '!' but works on floating stack

F @ Same as '@' but works on floating stack

F ' Same as ' ! ' but works on floating stack

F- Same as ' ' but works on floating stack

F+ Same as '+' but works on floating stack

F* Same as '*' but works on floating stack

F/ Same as '/' but works on floating stack

F= Same as '=' but works on floating stack

F > Same as '> but works on floating stack

F < Same as '< ' but works on floating stack

F. Prints the first number on the floating point stack

C64 I/O Commands

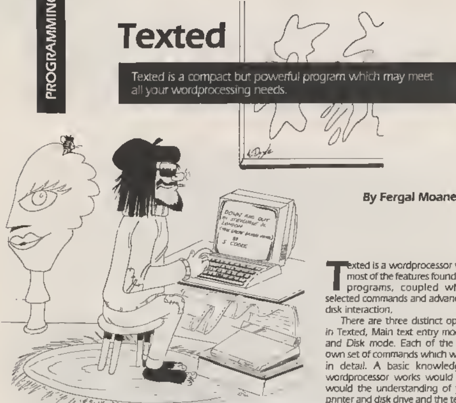
CHKIN
CHKOUT
CLOSE
OPEN
POS
SETNAM
TAB

FORTH command summary will be continued in the next issue

► ► ► ► ► ► ► ►

Texted

Texted is a compact but powerful program which may meet all your wordprocessing needs.



By Fergal Moane

Texted is a wordprocessor which provides most of the features found on commercial programs, coupled with easy Icon selected commands and advanced printer and disk interaction.

There are three distinct operating modes in Texted, Main text entry mode, Edit mode, and Disk mode. Each of the modes has its own set of commands which will be discussed in detail. A basic knowledge of how a wordprocessor works would be helpful, as would the understanding of your particular printer and disk drive and the terms associated with them.

Main text entry mode

This is the mode in which Texted starts. It is where text is entered and edited and where printer control commands are typed. This mode is signified by the flashing cursor on the workspace on the bottom of the screen. The icons are active in this mode

One key commands

Characters are typed at the position of the cursor up to a maximum of 80 characters at which the line is automatically entered into the main text.

RETURN enters the current line into the main body of text at the position of the Edit cursor.

LEFT ARROW moves the cursor to the end of the current line

UP ARROW enters the Edit mode

LEFT AND RIGHT CURSOR KEYS move the cursor over the type line.

HOME redraws the screen without corrupting the text in case of accidents

DELETE erases the character to the left of the cursor

INSERT has no function as in the Commodore screen editor, as text can be inserted by moving the cursor to the appropriate place and typing the text. It will be automatically inserted and the other characters moved up.

Icon controlled commands

Icons are activated by using a joystick in port 2 and moving the pointer to the appropriate icon and pressing fire. The selected icon will be highlighted

ICON 1 : Printer: Prints the text contained in the main body of text at the top of the screen, interpreting printer control commands as it meets them.

ICON 2 : Help: Calls up the Help screens containing a summary of commands and displaying the disk error channel.

ICON 3 : Pack: A space saver which packs your text into the optimum amount of space without harming printer control commands, which must be entered on a separate line.

ICON 4 : Key Beep: Switches the Beep sound on and off on all keys.

ICON 5 : Disk Goes into Disk mode where text is loaded and saved

ICON 6 : Edit Enters the edit mode, equivalent to pressing left arrow.

ICON 7 : Main Enters the default text entry mode

Printer control commands

All printer control commands are entered in the text entry mode as part of the main text, but preceded by an astensk (*) and entered on a separate line

The question of printer compatibility is a problem, so I have opted for conformation with the Epson standard (Star, Citizen etc.) while maintaining some commands for those with Commodore's own printers The commands are completely printer dependant and so will not work with printers which do not support this feature. Texted outputs to a Device 4 printer on the Serial bus, the normal Commodore convention Note that those commands that contain a numeric parameter must have the number within the range indicated in the brackets

Printer control summary

***colour(0-7)**	Prints in the specified colour (Star LC-10 etc)
***nlq**	Prints in Near Letter Quality
***draft**	Prints in the faster default draft quality
***italic**	Prints subsequent text in italics
***italic off**	Switches off the above, printing upright letters
***emphasized**	Adds emphasis to letters by thickening lines
***emphasized off**	Switches the above off
***doublestrike**	Produces dark text by printing twice
***doublestrike off**	Switches the above off
***reverse**	Prints white characters on black
***reverse off**	Prints normal black characters
***underline**	Underlines all subsequent characters
***underline off**	Switches the above off
***superscript**	Prints all subsequent characters as superscripts
***subscript**	Prints all subsequent characters as subscripts
***script off**	Switches the above two modes off
***expanded**	Expands text to double width
***expanded off**	Prints normal size text
***pica**	Prints in the default pica pitch
***elite**	Prints in the more compact elite pitch
***proportional**	Prints in proportional text
***proportional off**	Prints text with fixed spacing between letters
***double**	Prints characters at double width and height
***normal**	Prints at normal width and height
***bottom margin**	Sets a bottom margin of six lines on every page
***form feed**	Forced page, feeding to next page
***left margin(0-50)**	Sets a left margin of the indicated width
***right margin(0-50)**	Sets a right margin of the indicated width
***left justify**	Aligns the text with the left margin
***right justify**	Aligns the text with the right margin
***center**	centres the text between the two margins
***reset**	Resets printer to power-up state, clearing buffer

Edit mode

The edit mode is where major deletion and insertion takes place The current document is stored in the Yellow area at the top of the screen. The position of current operations is given by > This is where the next line to be typed is inserted, the line to be deleted is designated etc.

One key commands

RETURN exits to previous mode

DELETE removes the 80 column line beneath the edit cursor

C copies the line beneath the cursor down to the workspace to be edited

P sends text and embedded commands to printer

D goes to disk mode

F formats/packs text

UP AND DOWN CURSOR KEYS move the edit cursor up or down

Disk mode

[illegible]

[illegible]

Loading outside the menu

[illegible]

Extractor

Give your sprite collection a boost, by extracting sprites from other programs with this helpful utility.

By Neil Higgins

When used correctly sprites can, and often do, play a major part in most programs, whether they are used to animate a man in a game or just a simple pointer in a utility. This program will let you search through the whole of memory for sprites. Upon finding them, you can save them, watch the animations, move them around and copy them from one sprite position to another. As you can imagine, this is a very handy utility for all you budding sprite animators.

There are two versions of the sprite extractor, one sits in high memory from 49152 ($C000) to 53096 ($CF68) and the other in low memory from 20480 ($5000) to 24424 ($5F68). The reason for this, is that the sprites you wish to examine may sit in the same memory occupied by the extractor, if this is the case then you can load up the other version and avoid any clashes with sprite data

Using Extractor

First of all, load up a program that contains some of your favourite sprites or animation sequences, then if you have a reset switch, reset the computer then load up one of the extractors.

Alternatively, for people without a reset switch, if you load up the program to examine, but don't RUN it, you can then load the extractor. Most sprites should still be intact using this method, but for best results (as some programmers compact memory used by sprites), my advice is to purchase a reset switch. They can be bought quite cheaply now, and are invaluable when using the C64.

Okay, after starting the extractor you will be in view mode, this is where you locate the sprites and set the colours etc. On the right side of the screen you will see a menu with functions, most of these should be quite clear and you can experiment to see their actions. Displayed at the bottom of the screen are the current sprites pointer (0-255), the current bank (0-3), and the sprite data's address in decimal and hexadecimal. Some of the functions need further explanation; if you press 'C' to copy you will be prompted for the pointer number (0-255) and the bank number (0-3), please note that this copies the current sprite, and that there is no checking of where you wish to copy to, so be careful you don't overwrite the extractor! Press Run/Stop during any input to exit. Two extra functions not on the menu are:- key 'R' toggles repeat keys and Run/Stop exits to Basic.

Animation mode

If you find a sequence of sprites that you wish to animate, then press key 'A' in view mode and you will be asked 'Copy 30 sprites for animation? y/n'. If you are using the animation facility for the first time then press 'Y' – this will copy 30 sprites starting from the current one to sprite pointer 128 ($2000) in bank 0. The reason that the sprites need to be copied is because of the way the C64 loooks at its current bank for all sprites, re-defined characters and screen memory, so without using interrupts and bank switching, we need to copy any sprites we want to animate, into the default bank 0 For further information, consult the Programmers Reference Manual

You should now be in animation mode, where again the menu is on the right. The first sprite of your copied sequence is in the window, and if you plug a joystick into port 2 you will be able to move it around. Pressing key 'S' will alter the speed that the sprite moves. The number of frames in the sequence can be set by pressing 'F', and the speed of the animations by pressing 'A'. The animation can be started and stopped by pressing the joystick fire button. Note that you must move the sprite around to see it being animated

Certain keys also dictate the direction in which the animation is turned on

This means that animations will be turned on only when it is moving in the specified direction. Press 'X' to turn on in all directions. Experiment and you should get to grips with using the animation facility.

Sprite storage

To load or save sprites, press key 'D' in view mode. A menu will be displayed and it should be self explanatory. All sprites are saved from the current bank as in view mode. If you wished to save just one sprite, let's say the one at pointer 100, then you would enter 100 for both prompts (save from, to). All sprites are loaded into the address from which they were saved, and if you have a decent sprite editor it should allow you to load in sprites saved with the extractor, you can then of course re-edit them.

Sample sprites

For those of you who wish to practice using the extractor/animator, I have supplied two animated sprite sequences. The first is of an helicopter and the second a rather fat-looking cowboy, both are four frames long in multicolour. I will leave it up to you to set the colours. The sprites load into bank 0 at $1900 (POINTER 100), so if you go to pointer 100 in bank 0 you will see the first helicopter frame

When you have set some decent colours, press key 'A' and then key 'Y' to copy the sprites. You are now in animation mode, set the number of frames to 4, and plug a joystick into port 2, next set up the sprite movement and animation speeds, now move the sprite around and watch it being animated. To see the cowboy, go back to view mode, locate the first frame at pointer 104 and proceed as above.

Starting the extractor

Version LOW memory ($5000)
Start with SYS 20480

Version HIGH memory ($C000)
Start with SYS 49152

Loading the program

Sprite extractor comes in two files. To load the high version enter LOAD "EXTRACTOR.HI",8,1 and for the low memory version LOAD "EXTRACTOR.LO",8,1 That's all there is to it.

Windows 64

A handy little routine that allows you to specify your own adjustable screen windows

By Seb Reeve

This machine code routine provides fully adjustable windows on screen. These windows can be incorporated in any basic program to display messages, warnings or errors. The program itself was designed using the Basic Compiler by Paul Williams in the March/April Issue of the magazine. The Basic program that it was compiled from is also on the disk and is an example of the compiler in action.

The code itself is located at 2117. With the compiler Run-Time core at 49152, this however should not affect any Basic programs resident in memory.

Loading the program

The program can be loaded as follows outside the menu

LOAD "WINDOW1",8,1
SYS 64738

LOAD "WINDOW2",8

An example Basic program is provided To run this, or any other Basic program, you will need to relocate Basic program space to 8192 ($2000).

This is achieved by:

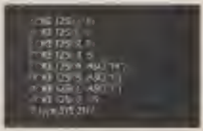
```
POKE 44,32:POKE 8192,0
LOAD "HI!"
RUN
```

The actual values needed by the window designer are:-

The following values should be POKED BEFORE typing SYS 2117 to open up a window:

POKE 12500, X value of top left corner
POKE 12501, Y value of top left corner
POKE 12502, L length in characters of window
POKE 12503, D depth in characters of window
POKE 12504+, ASCII Codes of letters of message to be contained within the window. The message should end with a '95' (A back arrow)

To print a window 10 chars long and 5 deep at character coordinates (10,6) with the message "Hi!" you would:



```
POKE 12500,10
POKE 12501,6
POKE 12502,10
POKE 12503,5
POKE 12504,ASC("H")
POKE 12505,ASC("I")
POKE 12506,ASC("!")
POKE 12507,95
Type SYS 2117
```

ZMON

We Commodore people tend to get a little obsessed with 6510 programming. We forget that lurking inside the C128 is a perfectly serviceable Z80. ZMON lets you get at it

by Neville Duguid

ZMON makes the Commodore 128's second microprocessor, the Z80, available to the built-in Machine Language Monitor. No longer is it necessary to import a machine-specific operating system like CP/M to try out the Z80. Just prefix your MONITOR command with 'Z' and ZMON will automatically invoke the Z80 to carry it out.

If you are unfamiliar with the 128's Machine Language Monitor, operating instructions may be found in Appendix J of the Commodore 128 System Guide. Except for 'A' (Assemble) and ';' (Modify Registers), ZMON's commands are similar for both processors. For example 'D 2AF' will disassemble the 21 bytes starting at address $02AF into 65XX-family instructions, whereas 'ZD 2AF' disassembles the same address range using Z80 mnemonics and Zilog-style syntax.

Those who have already mastered the 128's Machine Language Monitor will find ZMON transparent to use, except that this time, using ZMON, you are free to select the most appropriate microprocessor for the job at hand.

ZMON comprises two separate Machine Lanuage (ML) program files, "ZMON" (7 blocks) and "+ZMON.$7000" (14 blocks), both of which must be present on the same disk or tape.

To start ZMON, insert the disk containing both files or rewind your tape to the start of "ZMON".

If you have a Commodore 128D or are using a 1570 or 1571 disk drive:

```
BOOT "ZMON"
```

Users of disk drives that don't work with the 128's "BOOT" command - 1541's for example - will need to substitute the equivalent

```
BLOAD"ZMON":SYS DEC("1500")
```

Once an error-free load has occurred, beneath the copyright message and normal 8502 Registers, you will see 'ZR' (ZMON's "Z80 Registers" command) followed by the resulting output.

Striking contrasts

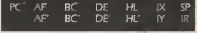
The most striking aspect of the display is the contrast between the two register sets. Not only has the Z80 more registers than the 8502, they are twice as big. That is only the tip of the iceberg.

Many of the Z80 registers on display, 'BC' for example, consist of two separate 8-bit registers ('B' and 'C' in this case) which, depending on the ML instruction accessing them, may be used either separately or together as a 16-bit pair. (When used in this way, the high-order byte is always the one on the left). Some of the registers, 'IX' for example, are strictly 16-bit, while one, 'IR', is nothing more than an eight-bit register and a hardware counter grouped together for convenience.

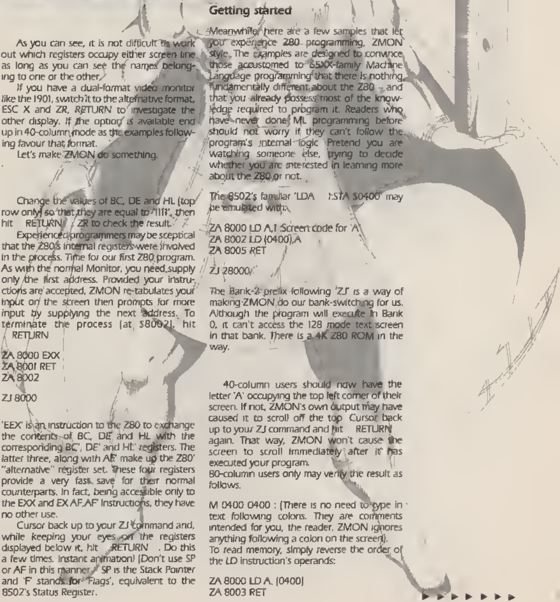
The ZR command's output will be tabulated according to the width of your display. If your current screen is 80-column, all the register settings will be on a single line beneath their corresponding register names.

In the 40-column version, half the names are missing and the register settings occupy two physical screen lines linked together as a single logical line. This arrangement allows 'Z;' to accept values from both lines without interference from a second header when the display is edited

Here is the format used to display the Z80's Registers on the 40-column screen



```
PC  AF   BC   DE   HL   IX  SP
    AF'  BC'  DE'  HL'  IY  IR
```



As you can see, it is not difficult to work out which registers occupy either screen line as long as you can see the names belonging to one or the other.

If you have a dual-format video monitor like the 1901, switch it to the alternative format, ESC X and ZR, RETURN to investigate the other display. If the option is available end up in 40-column mode as the examples following favour that format.

Let's make ZMON do something.

.

Change the values of BC, DE and HL (top row only) so that they are equal to '1111', then hit RETURN / ZR to check the result.

Experienced programmers may be sceptical that the Z80's internal registers were involved in the process. Time for our first Z80 program. As with the normal Monitor, you need supply only the first address. Provided your instructions are accepted, ZMON re-tabulates your input on the screen then prompts for more input by supplying the next address. To terminate the process (at $8002), hit RETURN

```
ZA 8000 EXX
ZA 8001 RET
ZA 8002

ZJ 8000
```

'EEX' is an instruction to the Z80 to exchange the contents of BC, DE and HL with the corresponding BC', DE' and HL' registers. The latter three, along with AF' make up the Z80' "alternative" register set. These four registers provide a very fast save for their normal counterparts. In fact, being accessible only to the EXX and EX AF,AF' instructions, they have no other use.

Cursor back up to your ZJ command and, while keeping your eyes on the registers displayed below it, hit RETURN . Do this a few times. Instant animation! (Don't use SP or AF in this manner. SP is the Stack Pointer and 'F' stands for 'Flags', equivalent to the 8502's Status Register.

Z80 programming is too vast a subject to be covered by a magazine article introducing a monitor program.

If you are serious about creating your own Z80 programs, you will need at least one good reference book on the subject. "Programming the Z80" by Rodney Zaks (SYBEX) is highly recommended.

Getting started

Meanwhile, here are a few samples that let you experience Z80 programming, ZMON style. The examples are designed to convince those accustomed to 65XX-family Machine Language programming that there is nothing fundamentally different about the Z80 - and that you already possess most of the knowledge required to program it. Readers who have never done ML programming before should not worry if they can't follow the program's internal logic Pretend you are watching someone else, trying to decide whether you are interested in learning more about the Z80 or not.

The 8502's familiar 'LDA 1:STA $0400' may be emulated with:

```
ZA 8000 LD A,1 Screen code for 'A'
ZA 8002 LD (0400),A
ZA 8005 RET

ZJ 28000
```

The Bank-2 prefix following 'ZJ' is a way of making ZMON do our bank-switching for us. Although the program will execute in Bank 0, it can't access the 128 mode text screen in that bank. There is a 4K Z80 ROM in the way.

40-column users should now have the letter 'A' occupying the top left corner of their screen. If not, ZMON's own output may have caused it to scroll off the top Cursor back up to your ZJ command and hit RETURN again. That way, ZMON won't cause the screen to scroll immediately after it has executed your program.

80-column users only may verify the result as follows.

M 0400 0400 : (There is no need to type in text following colons. They are comments intended for you, the reader. ZMON ignores anything following a colon on the screen).

To read memory, simply reverse the order of the LD instruction's operands:

```
ZA 8000 LD A, (0400)
ZA 8003 RET
```

▶ ▶ ▶ ▶ ▶ ▶ ▶

Type a character in the screen's 'HOME' position and instruct the Z80 to read it using one of the 'ZJ 28000' commands already on the screen.

Check the Accumulator (the 'A' in AF) in the Z80's register display It should contain the character's hexadecimal Screen Code.

80-column users should try this one instead:

```
ZA 8000 LD A,(0209) :InputBuffer+9
ZA 8003 RET

ZJ 28000:A
```

This time the tenth character in the input buffer – the one immediately following the colon attached to your 'ZJ' command – is retrieved. Try it with a few other characters The code is hex ASCII.

A more common method of accessing memory with the Z80 is to use the 16-bit register-pairs in parenthesis as pointers This may appear more complicated than the 8502's relatively straightforward approach, but there is an added bonus in mastering this style of notation 'C', a computer language very popular with many of today's top programmers, also uses this technique.

Here is a program that clears the 40-column screen using the Z80's DE register as a pointer to Screen Memory. As 16-bit INCs and DECs do not affect the Flags and there are no 16-bit 'CP' instructions in the Z80's instruction set, we have to improvise, using 'SBC' instead In this and subsequent examples, 40-column users will obtain results immediately affecting their screens 80-column users, on the other hand, will need the 'ZM' command to see the output of their programs. Perhaps they will be the first to realize why ZMON is called a "monitor".

```
ZA 8000 LD A,$20        Screen Code,
   SPC
ZA 8002 LD DE,$0400     :Starting address
ZA 8005 LD (DE),A       :Output a byte
ZA 8006 INC DE          :Next address
ZA 8007 LD HL,$07EB     :End address+1
ZA 800A SBC HL,DE       :Bytes remaining
ZA 800C JR NZ,$8005     :Loop till count
                        =0
ZA 800E RET

ZJ 28000
```

ZMON's Registers are displayed at the old cursor position immediately after the Z80 program terminates.

This program may be used to fill the screen with any character you choose. Just put the appropriate Screen Code in 'A' (the first two digits of AF) and skip the first LD instruction. ZJ 28002: (When trying different characters, include the colon so that ZMON will ignore previous characters already filling the screen).

If you forget the Bank 2 prefix, no output will reach the screen. The Z80 can't write "through" its ROM. This ROM appears at $0000-$0FFF in Z80 mode whenever bits 6 and 7 of the Configuration Register are clear (as occurs in Bank 0). In that condition the instruction sequence 'LD HL,$FF00:SET7,(HL)' may be used to switch the Z80 ROM out of what would otherwise be Bank 2 Prefixing the ZJ (or ZG) target address with '2' is an easier way to achieve the same result. Here is a way to confirm it:

```
ZM FF00 FF00
ZM 2FF00 2FF00
```

The only difference is that Bit 7 is set in Bank 2 Anything above $0FFF in Bank 0 will be available in both banks.

To see the Z80 ROM, 'ZD 0' and 'ZD' again until you have seen enough

Changing the border

Here's how to change the 40-column screen's border using the Z80:

```
ZA 9000 LD A,2          :Red
ZA 9002 LD BC,$D020     :Border
ZA 9005 OUT (C),A
ZA 9007 RET

ZJ 9000
```

80-column users:

FD020 : Only the right hand hex digit will be significant in this case – try editing the result.

Although your Z80 text book will probably describe IN and OUT in terms of eight-bit "port numbers", 'OUT (C),A' is really equivalent to 'OUT (BC),A' on the Commodore 128 which uses all 16-bits of its address bus to select the Input/Output (I/O) ports The instruction works exactly the same as 'LD (BC),A' except it accesses the I/O devices at $D000-$DFFF whereas most of the other Z80 instructions do not. At addresses outside that range, OUT has access to the same memory as LD.

To highlight the similarity of these two instructions, we will display the current character set on the screen using LD for the character codes and OUT for the colour.

```
ZA E010 LD A,7          :Yellow
AZ E012 LD HL,$0600     :Mid-screen
```

```
ZA E015 LD BC,$DA00 :Colour Memory
ZA E01B LD (HL),L   :Output a char
ZA E019 OUT (C),A   :Output colour
ZA E01B INC C       :Next address
ZA E01C INC L       :Next addr & char
ZA E01D JR NZ,$E018 :Loop while L    0
ZA E01F RET
```

```
ZJ 2E010
```

Unlike their 16-bit counterparts, the INC instructions used in this program affect the Flags, allowing us to use the low byte of an address pointer to terminate the loop when it "rolls over" to zero.

Color Memory may also be switched into the Z80's main memory at $1000-$13FF by resetting bit-0 of the Configuration Register. This happens automatically whenever a bank prefix higher than 3 (except 14) is used.

Advantages of the "memory mapped" colour table include use of normal instructions – you can cycle colours by INCing them, for example. Disadvantages include the hazards inherent in back-switching. Don't locate your program above 3FFF or below $1400 in Bank 15, for example. Program code between $1400 and $7FFF in Banks 6 and 10 will be in context with both Screen and Color Memory at the same time. Here is a way to demonstrate the presence of both in Bank 6. This time, be sure to include the colons.

```
ZF 60400 607E7 21: "!"
ZF 61000 613E7   3. Cyan
```

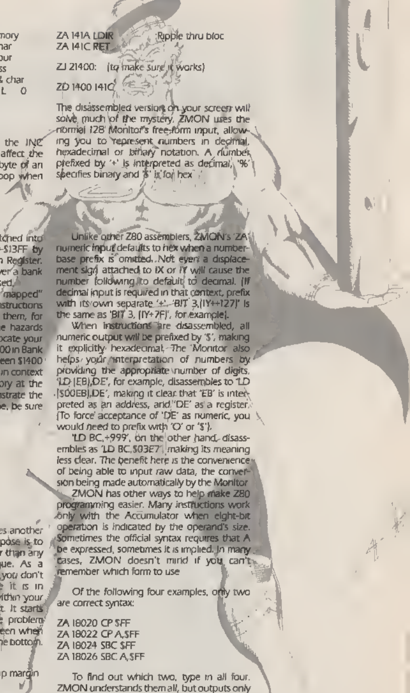
Clearing the screen

The following program demonstrates another way of clearing the screen. Its purpose is to illustrate ZMON's 'ZA' syntax rather than any specific Z80 programming techique. As a subroutine it is easy to use, even if you don't understand how it works. Once it is in memory, just 'CALL $1400' from within your own program (in Bank 2) to use it. It starts by homing the cursor – fixing the problem that occurs if ZMON scrolls the screen when we 'ZJ' with the cursor too close to the bottom.

```
ZA 1400 LD IX,EB       :Cursor x,y
ZA 1404 LD E, (IX-6    :Window, top margin
ZA 1407 LD D, (IX-+5)  :left margin
ZA 140A LD (EB),DE     :16-bit plot x,y
ZA 140E LDA,+32        :char =   SPACE
ZA 1410 LD HL,+1024    :Overlap source bloc
ZA 1413 LD DE,$401     :with DEstination
ZA 1416 LD BC,+999     :(bloc size)
ZA 1419 LD (HL),A      :Fill source byte
ZA 141A LDIR           :Ripple thru bloc
ZA 141C RET
```

```
ZJ 21400:   (to make sure it works)
```

```
ZD 1400 141C
```



The disassembled version on your screen will solve much of the mystery. ZMON uses the normal 128' Monitor's free-form input, allowing you to represent numbers in decimal, hexadecimal or binary notation. A number prefixed by '+' is interpreted as decimal, '%' specifies binary and '$' is for hex.

Unlike other Z80 assemblers, ZMON's 'ZA' numeric input defaults to hex when a number-base prefix is omitted. Not even a displacement sign attached to IX or IY will cause the number following to default to decimal. (If decimal input is required in that context, prefix with its own separate '+'. 'BIT 3,(IY++127)' is the same as 'BIT 3, (IY+7F)', for example).

When instructions are disassembled, all numeric output will be prefixed by '$', making it explicitly hexadecimal. The Monitor also helps your interpretation of numbers by providing the appropriate number of digits. 'LD (EB),DE', for example, disassembles to 'LD ($00EB),DE', making it clear that 'EB' is interpreted as an address, and 'DE' as a register. (To force acceptance of 'DE' as numeric, you would need to prefix with '0' or '$').

'LD BC,+999', on the other hand, disassembles as 'LD BC,$03E7', making its meaning less clear. The benefit here is the convenience of being able to input raw data, the conversion being made automatically by the Monitor.

ZMON has other ways to help make Z80 programming easier. Many instructions work only with the Accumulator when eight-bit operation is indicated by the operand's size. Sometimes the official syntax requires that A be expressed, sometimes it is implied. In many cases, ZMON doesn't mind if you can't remember which form to use.

Of the following four examples, only two are correct syntax:

```
ZA 18020 CP $FF
ZA 18022 CP A,$FF
ZA 18024 SBC $FF
ZA 18026 SBC A,$FF
```

To find out which two, type in all four. ZMON understands them all, but outputs only in standard form.

To be continued

Cribbage Master

As far as we know, this is a C64 first. A must for beginners and experienced cribbage players alike.

By Paul Schofield

It's time to roll out the green baize, because your C64 is waiting to challenge you at Cribbage I am sure most readers are familiar with at least the basic rules of Cribbage, but for the complete novice here's a short introduction to get you started.

A brief introduction

Cribbage is a card game played either between two players or four players, playing as two pairs. The program does not support the doubles version, so this complication can be ignored. The object of the game is to score 121 points before your opponent. Traditionally the score is indicated on a pegboard, but if you find this confusing, the computer also shows the score in digital format.
You can score points in two ways:

- scoring combinations in your hand
- during pegging

Before looking in detail at the scoring of points, we will look at the format of a game. At the start the two players cut for deal. The Ace counts as a low card and the player who cuts the highest card deals first The dealer then deals a hand of six cards to both players. Each player examines his hand to try to calculate the highest score which can be achieved with any four of the cards and the other two are discarded. The four discarded cards form a third hand called the box, which becomes the second hand of the dealer. When both players have discarded, the non-dealer cuts the pack and the top card is turned face up. This will be used later with the cards in the hands, but if it is a Jack the dealer scores 2 points. This is termed "2 for his heels"

The next stage of the game is the pegging The players take it in turn to lay cards starting with the non-dealer. Points can be scored during pegging as follows:

* If the total sum of the points of all cards played = (all picture cards count 10), the player who played the last card scores 2 points

* If the last two or more cards played have the same rank, the player who played the last card scores 2 points for each pair (e g three 8's = 6 points)

* If the last n cards are consecutive, the player who played the last card scores n points.

* If the total sum of points of all cards played = 31, the player who played the last scores 2 points

This process continues, until the total reaches 31 or no player holds a card, which can increase the current total without exceeeding 31. When 31 is not reached the last player to lay a card scores I point, the cards are then turned face down and the process continues until both players have played all their cards.

This probably sounds rather confusing, but the program won't let you make illegal plays and you should soon get the hang of it. For now, one example will give you a general idea of how it works:

PLAYER A holds 9-8-8-7, which is a good hand as you will see later.
PLAYER B holds 8-7-6-A, which is quite similar to make things interesting.

B was dealer, so player A starts the pegging. A plays an 8 and hopes B has one and will try to make a pair in which case they could then lay their second 8 and score 6 (for 3 pairs).

B does not want to risk this, particularly as by playing the 7 he can score 2 for 15 points. They do this instead.

A could now play the 7 to score two for a pair, but prefers to play the 9 to make the score 24. The last three cards are 8-7-9 scoring 3 for a run.

This turns out to be exactly what B was hoping for. B now plays his 6 taking the score to 30 and scores 4 points for a run.

Player A cannot go as his lowest card is the 7, but B has an Ace (value 1) to make the total 31 and get a further two points

The cards played are now turned face down and it is A's turn to play again. A's misfortune continues as, whichever of the remaining two cards lead will allow B to score a further 2 points, before A gets a consolation point for last card.

In this example B pegged 10 points to A's 4, which can have a dramatic effect in a close game Such high scoring during pegging normally only occurs when the two players hold similar hands at it is not uncommon for the players to score only 2 or 3 points between them.

At the end of the pegging, the hands are scored. The non-dealer goes first, followed by the dealer and then the dealer's box This order is important, because the game ends as soon as one player's score reaches 121 and unscored points in the opponent's hand or box count for nothing.

When scoring the hands, the 4 cards retained plus the card turned up earlier are used. Points are scored in a similar manner to pegging:

* Each combination adding up to 15 scores 2 points
* Each pair scores 2 points
* Each run of n cards scores n points

There are also a few differences'

* A Jack of the same suit as the turn-up card scores 1 point (1 for His Nob)
* All of the original 4 cards of the same suit scores 4 for a flush (if the turn up card is also of this suit scores 5)
* No points for combinations of 31

Consider the hand of player A when the card turned up is the Queen of clubs. The scoring hand is 9D-8H-8S-7S plus QC. The hand is now scored as follows:

Cards Used	Points	Total	How it's said
7S-8S	2	2	15 for 2
7S-8H	2	4	15 for 4
8S-8H	2	6	and 2 for a pair is 6
7S-8S-9D	3	9	and 3 for a run is 9
7S-8H-9D	3	12	and 3 for a run is 12

This is an above average score for a hand, 7 or 8 points being typical. Scores for the box are usually much lower, so you should be pleased with anything over 4 points. Do not worry about miscounting your hand. The computer will tot up the points accurately for you.

When all hands have been scored, the dealer changes and the next hand is dealt. You know enough about Cribbage to challenge the Cribbage Master. If you watch what the program does with good hands you should soon pick up a few more tips. Don't take so much notice of what it does with poor hands, it seems to have acquired the human trait of losing interest when dealt poor cards.

Playing cribbage master

If you want to [illegible] Cribbage Master [illegible]

[illegible]

[illegible]

[illegible] To [illegible] program [illegible] LOAD [illegible] RUN [illegible] After entering your name [illegible] joystick in [illegible] LEFT [illegible] RIGHT [illegible]

ON THE DISK

Mini-spread 128

How to use last issue's spreadsheed (Oct/Sept CDU) to set up formulas

By Kevin Blight

Using Formulas

Mini Spread has a number of functions which can be used within formulas, a formula is just a string of these functions with cell references included. A full description of each command is given below. In each function AA02 is used to sndw the first cell and AE02 is used to show the last cell.

TOT: This function is used to total either a row or a column, the function is written as follows:-
TOT AA02-AE02

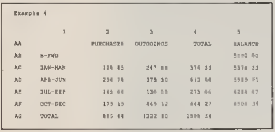
MAX: This function is used to find the maximum value in a row or column, the function is written as follows :-
MAX AA02-AE02

MIN: This function is used to find the minimum value in a row or column, the function is written as follows:-
MIN AA02-AE02

AVG: This function will find the average value in a range of cells, the function is written as follows:-
AVG AA02-AE02

BAL: This function is used to find the balance of a row, the value of the above formula cell is added to the result found. The function is written as follows:-
BAL AA02-AE02

An example of the purpose of this function will help to explain its importance.



Example 4

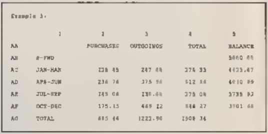
	1	2	3	4	5
AA		PURCHASES	OUTGOINGS	TOTAL	BALANCE
AB	B-FWD				5000.00
AC	JAN-MAR	128.45	247.88	376.33	5376.33
AD	APR-JUN	234.78	377.90	612.68	5989.01
AE	JUL-SEP	145.06	130.00	275.06	6264.07
AF	OCT-DEC	175.15	469.12	644.27	6908.34
AG	TOTAL	683.44	1222.90	1906.34	

Formulas for this example .

AC : 04 : TOT AC02-AC03
AC : 05 : BAL AC02-AC03
AD : 04 : TOT AD02-AD03
AD : 05 . BAL AD02-AD03
AE : 04 · TOT AE02-AE03
AE : 05 : BAL AE02-AE03
AF : 04 : TOT AF02-AF03
AF 05 : BAL AF02 AF03
AG : 02 : TOT AB02-AE02
AG · 03 : TOT AB03-AE03
AG : 04 : TOT AB04-AE04

The four formulas in column 05 provide the balance by adding the cell above the total of columns 02-03.

SUB: This function is used in a similar way to BAL, except that the above cell is subtracted from the total of the row in a similar way to a bank statement, where the balance is reduced each time. The function is written as follows:-
SUB7 AA02-AE02
An example will help to make this clearer.



Example 3:

	1	2	3	4	5
AA		PURCHASES	OUTGOINGS	TOTAL	BALANCE
AB	B-FWD				5000.00
AC	JAN-MAR	128.45	247.88	376.33	4623.67
AD	APR-JUN	234.78	377.90	612.68	4010.99
AE	JUL-SEP	145.06	130.00	275.06	3735.93
AF	OCT-DEC	175.15	469.12	644.27	3091.66
AG	TOTAL	683.44	1222.90	1906.34	

Formulas for this example:-

AC : 04 : TOT AC02-AC03
AC : 05 : SUB AC02-AC03
AD : 04 : TOT AD02-AD03
AD : 05 : SUB AD02-AD03
AE : 04 : TOT AE02-AE03
AE : 05 . SUB AE02-AE03
AF · 04 : TOT AF02-AF03
AF : 05 : SUB AF02-AF03
AG : 02 . TOT AB02-AE02
AG : 03 : TOT AB03-AE03
AG : 04 : TOT AB03-AE04

As you should be able to see, this has a similar function to BAL except that the balance is subtracted from the above cell instead of adding to it.

COU: This function is used to count the number of entries in a row or column Only cells with a value larger than 0.00 are counted. The formula is written as follows:-
COU AA01-AA03
When you come to use Mini Spread to set up your own spreadsheets, you will have to write your own formulas. I have tried in the previous examples to show how formulas can be set up. If you load the file PROFIT.MSD from the disk you will see this is a large sheet showing the profit and loss for a fictitious company There are a large number of formulas on this sheet using most of the functions of Mini Spread

One useful formula is one which enables you to remove the VAT on an item, this is done by using the formula:-

'(AA01) / 23 * 3' - Where AA01 is replaced with the cell to remove the VAT from.

The example sheet occupies 14 columns so you will not be able to print it out unless you have SIDEWAYS 128, but you can print it out in a number of pieces

I hope you have as much fun using Mini Spread as I had in writing it

Summary of commands

C: Copy [illegible]
[illegible] formula in cell
F. Enter formula in cell
H or HELP: Display help screen.
I: Insert / [illegible]
J: Justify [illegible]
L: Load new sheet
M: Display free memory
P: Print [illegible]
Q: Quit Mini Spread
R: [illegible]
S: Save current sheet
T: Enter text in cell
U: [illegible]
V: Enter value in cell
W: Change [illegible]
(Column [illegible])
X: [illegible]
[illegible]
[illegible]
Z: [illegible]

CURSOR UP: Up 1 row
CURSOR DOWN: Down 1 row
CONTROL [illegible]
CONTROL [illegible]
CONTROL UP: Up 1 screen
CONTROL DOWN: Down 1 screen

Array storage

There are three arrays used in this program to store the data in the sheet, these are CE$(), CE%() and CE(). CE$() is used to store text and formulas. CE() is used to store values and values from calculated formulas. CE%() is used as a flag to show the type of entry in a cell. The values of the flag are as follows:-

0: Text cell.
1: Value cell
2. Formula cell.

These arrays are two-dimensional, dimensioned to CE$(100,25) The first reference is used for row and the second is used for column.

Variable/Function

FNM():Function to display available memory for data.
. FRE(0) - 2000 (For program variables)
CE$() : Text / formula entries.
CE%(): Cell type flag
CE(): Value entries
F$(): Used to split formulas up
MR%. Maximum row used
MC% Maximum column used
SC%: Sheet calculated flag. 1=Yes / 0=No
R%: Current row at top of screen.
C%: Current column at top of screen.
LC$: Lowercase switch.
UC$: Upper case switch
PU$: Print using string.
SP$: String of spaces.
EC$ String of 11 spaces for empty cell
E1$-E9$: Error messages.
FI$ Current file name

Loading the program

Mini Spread 128 is written in compiled Basic. There are three programs which make up Mini Spread, these are as follows:-

MINI SPREAD 1:	Title program to BLOAD 'MINI SPREAD 2 and run 'MINI SPREAD 3
MINI SPREAD 2	Binary file for machine code input routine.
MINI SPREAD 3:	Actual compiled BASIC program

Oblivion!

Yet again, Terra is threatened by the monstrous Jandoids. Your job? To stop them

By Martin Jones

Masters of weather control, the beastly Jandoids have at last perfected their parasite power pods. Their purpose is to drain gigatons of energy from Terra's climatic systems, thus causing the onset of a new Ice Age, with disastrous consequences for the planet and its inhabitants.

Can they be stopped? The UN World Peace Council seems to think so. They've selected you, the planet's finest, to pilot a flimsy craft, which is, however, the best that Terra's depleted industry can come up with, to enter the Jandoid energy base and eliminate the threat.

Your main directive is simple. Manoeuvre through the caverns inside the Jandoid asteroid base and zap the power pods which are draining the Earth. Many perils must be avoided Any contact with the walls will bring instant death. To make matters worst, you will have to dodge enemy gun emplacements armed with precision-guided munitions

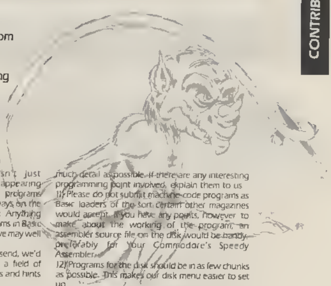
You will have to complete the full nine levels of the game to definitively eliminate the threat. Players will need a joystick in Port 2 to move up, down, left and right and fire. ■

Loading the Program

To load the program outside the usual menu, enter LOAD "OBLIVION", 8 and RUN

Contributions

Written some programs?
Got some programming wisdom
to pass on? Or do you want
to write about your own
fields of interest? We're waiting
for your contributions.



Commodore Disk User doesn't just offer you the chance of appearing in print, but of putting your programs on our disk for all to admire We're always on the lookout for new programs for the disk Anything goes, utilities games or business programs in Basic or machine code - if we think it s good, we may well publish it

Even if you haven't got a program to send, we'd love to pick your brains If you have a field of expertise you'd like to explain or any tips and hints of interest to disk users, send them in

But how do you go about preparing a submission? Just follow the guidelines and all should go well You don't have to be a great novelist to contribute, but if you follow our simple rules then it will make our job a lot easier

1) If possible all material sent to the magazine should be typed or printed out on a computer printer

2) All text should be double-spaced, i e there should be a blank line between each line of text You should also leave a margin of at least 10 characters on each side of the text.

3) On the first page you should put the following
Name of the article
Machine that it is for (C64/128)
Any extras required - disk, printer, add-ons etc
Your name
Your address
Your telephone number

4) The top of every page should have the following information on it.
Abbreviation of the article title
Your name
The page number

For example, suppose you had submitted a piece on C64 3D graphics You should put something like this at the head of the page
3D/G Brown/1

5) Please make sure that you do not make any additional marks on your text, especially underlining

6) Try to write in clear concise English Your contribution does not have to be a great work of literature, but it must be comprehensible

7) On the bottom of each page you should put the word MORE if there are more pages to the article or ENDS if it is the last page

8) If possible, enclose a listing of all programs

9) Use a paperclip to hold the pages together Do not staple them

10) When submitting programs for the disk, submitting the program alone is not enough Please tell us how to load, run and use it, preferably in as much detail as possible. If there are any interesting programming point involved, explain them to us

11) Please do not submit machine-code programs as Basic loaders of the sort certain other magazines would accept. If you have any points, however to make about the working of the program, an assembler source file on the disk would be handy, preferably for Your Commodore's Speedy Assembler.

12) Programs for the disk should be in as few chunks as possible. This makes our disk menu easier to set up

13) Programs under 10 lines can be included in the text. If your program is longer than this it must be on a disk.

14) If your article needs any artwork, then supply clear examples of what you want. We don't expect you to be an artist, but we do need to see what is required

15) Photos, if necessary, must be either black and white prints or colour slides We can take shots ourselves, so don't worry about this too much

16) Submissions of any length are welcome A five-line routine may be just as welcome as a six-part series of 2000-word articles.

17) Payment can vary from £50 for a very short routine to £700 for a large program published in installments, and depends on quite a number of factors, such as complexity and presentation of program For articles, the number of magazine pages taken up is the salient factor

18) All payments are made in the month that the magazine containing your article has appeared in print

19) If we do find your submission suitable for inclusion in the magazine, we will write to you giving the terms of publication, the rate of payment, and an agreement form. Prompt return of this form will allow us to use your program as soon as possible

20) If you want the program to be returned to you, should we find it suitable for publication, then you should enclose a stamped addressed envelope

21) If you use a wordprocessor, then enclose a copy of your text on the disk and state clearly which wordprocessor you use

22) Send your programs and articles to
Commodore Disk User
Submissions
1 Golden Square
London W1R 3AB

23) Commodore Disk User cannot accept any liability for items sent to the magazine

Bumper Sticker Maker

Is the C64 at last becoming the right machine for advanced applications? Myron Patch investigates

One frustrating thing about the C64 is that it never seems to come of age as a machine for advanced applications such as weather forecasting, the design of precision-guided munitions, or the analysis of Voyager space-probe data. Up until now, organisations such as the British Weather Centre and NASA have had to rely on over-priced and bulky Cray so-called 'supercomputers'. The price to the taxpayer is incalculable, and the attendant social costs, in that computer staff cannot take their work home and are therefore required to subsist on a diet of inferior office coffee, are beyond any sane estimate

As we all know, life could be much simpler if only the software existed to take advantage of the C64's awesome processing capacity Some progress has been made in the games sector, but it is astonishing how many worthless and trivial applications, such as. databases; spreadsheets; and wordprocessors, have been implemented on the poor old Commodore.

This seems destined to change. Walrusoft's *Bumper Sticker Maker*, marketed in the UK by Financial Systems Software, may well be a decisive intervention in the advanced applications market.

...and why not?

In a burst of sparkling insight, the minds behind BSM have analysed the gaps in the current automotive aesthetic. Once seen as the very quintessence of evanescence in more sedate times, these days the automobile bumper sticker has acquired a certain stodgy permanence. This can largely be seen as due to far-reaching improvements in adhesive technology, not to speak of giant leaps in laminated thin-film polymers.

BSM, however, celebrates impermanence itself. With BSM, the enthralling prospect now exists of changing your sticker every day in a sort of 0.0000115 frames per second animation sequence. Cheap, biodegradable materials, in the form of simple printer-loadable paper labels are used throughout, and the essential imagery sits on disk, waiting for a new day's modificaton. Vorschprung durch Technik!

Shattering simplicity

Realising that software users have no wish to be burdened with an excess of confusing features, Walrusoft has taken the refreshing course with this program of providing next to none.

For text, the would-be bumper designer is provided with a variety of predesigned fonts. There is no attempt to befuddle the user by allowing them to design their own. This otherwise impeccable scheme is *slightly* flawed, because it *is* possible to double height and/ or width, and add underlining.

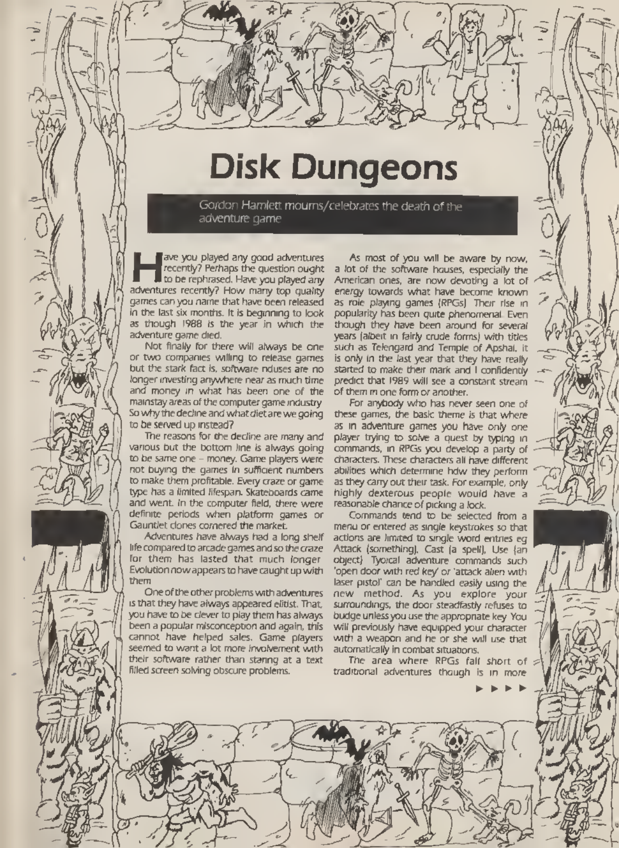
Besides text, there is only one other class of semiological construct available in BSM, and that is the 'icon', a choice of terms which surely implies a passionate analytical involvement with the works of Althusser and Barthes. Icon construction is a pleasingly simple process. Fundamental elements are confined to lines, boxes and circles.

Freehand drawing with BSM enforces a refreshing discipline. Erasure has been made deliberately tricky, thus enforcing use of a studied and precise technique. Too many art packages pamper the user, allowing them, for example, to take back commands they feel to be wrongly entered, surely not an encouragement to either spontaneity or resolve. Not so BSM – this is not a piece of software for the fainthearted

Conclusion – for now...

I could continue to sing the praises of this extraordinary package, but I think that I have told you enough to give you an impression. The writing is now on the wall for those software houses who persist in following the now-obsolete 'user-friendly' approach. ■

At a glance
Title: Bumper Sticker Maker
Supplier: Financial Systems Software
Price: £24.95



Disk Dungeons

Gordon Hamlett mourns/celebrates the death of the adventure game

Have you played any good adventures recently? Perhaps the question ought to be rephrased. Have you played any adventures recently? How many top quality games can you name that have been released in the last six months. It is beginning to look as though 1988 is the year in which the adventure game died.

Not finally for there will always be one or two companies willing to release games but the stark fact is, software nduses are no longer investing anywhere near as much time and money in what has been one of the mainstay areas of the computer game industry So why the decline and what diet are we going to be served up instead?

The reasons for the decline are many and various but the bottom line is always going to be same one – money. Game players were not buying the games in sufficient numbers to make them profitable. Every craze or game type has a limited lifespan. Skateboards came and went. In the computer field, there were definite periods when platform games or Gauntlet clones cornered the market.

Adventures have always had a long shelf life compared to arcade games and so the craze for them has lasted that much longer. Evolution now appears to have caught up with them

One of the other problems with adventures is that they have always appeared elitist. That, you have to be clever to play them has always been a popular misconception and again, this cannot have helped sales. Game players seemed to want a lot more involvement with their software rather than staring at a text filled screen solving obscure problems.

As most of you will be aware by now, a lot of the software houses, especially the American ones, are now devoting a lot of energy towards what have become known as role playing games (RPGs) Their rise in popularity has been quite phenomenal. Even though they have been around for several years (albeit in fairly crude forms) with titles such as Telengard and Temple of Apshai, it is only in the last year that they have really started to make their mark and I confidently predict that 1989 will see a constant stream of them in one form or another.

For anybody who has never seen one of these games, the basic theme is that where as in adventure games you have only one player trying to solve a quest by typing in commands, in RPGs you develop a party of characters. These characters all have different abilities which determine hdw they perform as they carry out their task. For example, only highly dexterous people would have a reasonable chance of picking a lock.

Commands tend to be selected from a menu or entered as single keystrokes so that actions are limited to single word entries eg Attack (something), Cast (a spell), Use (an object) Tyoical adventure commands such 'open door with red key' or 'attack alien with laser pistol' can be handled easily using the new method. As you explore your surroundings, the door steadfastly refuses to budge unless you use the appropriate key You will previously have equipped your character with a weapon and he or she will use that automatically in combat situations.

The area where RPGs fall short of traditional adventures though is in more

▶ ▶ ▶ ▶

complex problem solving There are two reasons for this. One, the command structure is not complex enough to do the sort of things a good parser can handle although this is something that is improving all the time Most command structures can be broken down into 'do X with Y' What is more worrying though and the area where traditional adventurers have most ground for complaint is that the storyline is very thin and does not require any major attempt at problem solving.

There are other aspects of RPGs that only became apparent from talking to other players. Unlike adventures, each game develops differently as there is a considerable random element involved as to what monsters you meet and treasure you find. I've noticed people in the CDU office new to RPGs having quite animated discussions as to whether monks were better than hunters, which spells worked well in which situations and so on This is something that never happens with adventures. You can only discuss these games at risk of ruining the plot.

Another feature of the games is that there is always something to do. Unlike adventures where you can come to a total dead end that won't change until you solve a particular problem, you can always explore the land further looking for additional clues. You know that they are there somewhere, it is just a matter of finding them. Unlike adventures where you need the exact phrase to proceed

One old age pensioner wrote in to me to say now much he enjoyed games of this type. He felt that there was plenty for him to do and the game progressed at just the right pace – his reactions weren't good enought for arcade games but ne enjoyed the challenge of RPGs – the exploration combined with strategy elements. Even if not all game players are that old, it is worth while remembering that games players in the States have a much older average age than their UK counterparts (25-35 rather than 12-18) and that nearly all RPG software is currently being written in the States and thus aimed at an older audience

As it stands at the moment, a typical RPG scenario consists of little more than wandering round exploring everything and picking up a couple of key items before you are strong enough to take on and destroy the ultimate evil. Detractors should remember though that the stories and problems in the early adventures strained credulity to the limits and this in time is something that I think will improve in leaps and bounds.

To compensate for these misgivings, there are several aspects of RPGs that are absent from traditional adventures. As most of the games are combat orientated, meaning you kill monsters to gain treasure and experience, combat sequences can develop into some very good exercises in strategy. With more than one character to think about, you must work out how best to optimise your forces Who goes in the front line? Snould I use offensive or defensive spells? How do I deal with enemies that are some way distant but who are still casting spells at me? Do my fighters shoot arrows at them leaving them vulnerable to the monsters in close contact with them, or do I use up some valuable magic? This strategical element can also be increased several fold by having the characters able to move as well as hit.

There have been many attempts to incorporate magic into adventures but none of them have worked particularly well. For anyone fancying themselves as a latter day Merlin, most RPGs offer a comprehensive set of spells with the added bonus that the use of magic fits in easily with the game system rather than sitting uncomfortably on top of it.

One final thought, if you do enjoy games of this type, they do offer fantastic value for money with games taking maybe hundreds of hours to complete. I consider myself a great adventure fan but I confess that RPGs have kept me burning the midnight oil much more than any other game did.

▶ ▶ ▶

Hints and Tips

Bard's Tale II

A useful combination of Bard songs in a dungeon is the Rhyme of Duotime as you explore in order to restore your spellcaster's points followed by Sanctuary Score on the first round of combat to lower the armour class of your party.

Don't be afraid of using magic items when you find them. There are several versions of most of them and you are likely to find that they are not particularly helpful later on in the game.

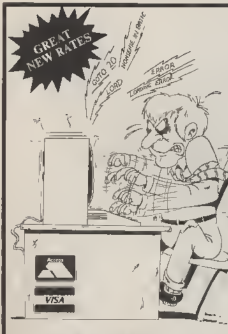
The Sorceror's Mind Jab spell is an effective way of dealing with solitary enemy spellcasters 30-50 feet away. It is also cheap at only three spell points. Similarly, Star Flare is the cheapest spell to use on a group of monster of low to medium level.

Trap Zapping chests is the best way of opening them even if your thief is well developed. It also proves cheaper in the long run as you don't have to use costly spells to cure someone if disarming the trap goes wrong.

Remember that any summoned creature in your party will always attack the first group of enemy monsters. If he is to do a lot of damage, put your fighters on to somebody else.

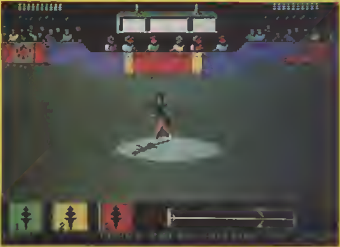
Useful spells to keep permanently cast are Magic Compass and Sorceror Sight.

Amongst the magic items, horns cause damage to a group of monsters and are particularly useful against creatures such as goblins who don't do any real damage but whose numbers make them annoying. Figurines summon characters to join your party (one use only). Make sure that you have a vacant slot for them. If you want to summon a new creature, check that the old one isn't carrying anything important before you dispell him.



The Games – Winter Edition

Tony Hetherington takes on the elements in Epyx's latest blockbuster.



Snow, ice, skates and skis combine in this the second compilations of winter sports in an Epyx game. Winter Games was my favourite ahead of World, California, Summer and Summer II Games but now it has some competition

The Games – Winter Edition has been launched as a result of Epyx's success in capturing the official Olympic license and will be followed by The Games – Summer Edition in time for the Seoul Olympics. The Winter edition launched in Summer and the Summer edition in Winter! This could only happen in the software business.

The game itself consists of seven events loaded in from a menu from which you can decide whether to practice or compete in one or more of the athletic tests that lie ahead. As in the other "games" games up to eight players can compete representing one of 16 countries.

As soon as you start playing the game you'll to notice some changes from the usual format as the events have become more involved. It's not just another in the series.

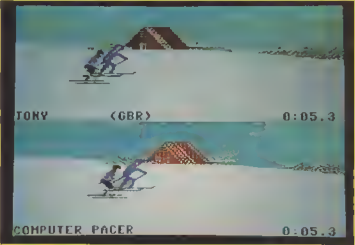
Before the events can begin the games are opened in a ceremony where a runner flanked by Mounties, runs up the steps to light the Olympic flame

The first event is a test of skill and nerve as you climb into a luge to hurtle down one of four tracks. The start is all important and you have 30 seconds to build up momentum by rocking the luge before setting off down the ramp that leads to the course with its high speed straights and hairpin bends. Now you must move the joystick to drive along the correct line through the track by keeping an eye on three gauges that show your position, where you're steering and the drift. If you get it right the time might be enough for a gold medal and even a world record.

Cross Country is a test of endurance in a sideways scrolling race against the clock and your opponents over one, two or five kilometre courses. It plays a little like the Biathlon in Winter Games but without the shooting which makes a simple case of keeping a good rhythm and enough speed to get you up the hills

Figure Skating makes a reappearance but this time it's a lot more involved as it's now a two part process. In the first part you must select your music from the seven styles that range from rock to jazz and then plan your program. To have a complete program you must perform each of the eight moves represented by icons once and at least ten moves in all However, you must also plan your moves so that they are performed in time with the music so gauges at the bottom of the screen make sure you know what you're doing.

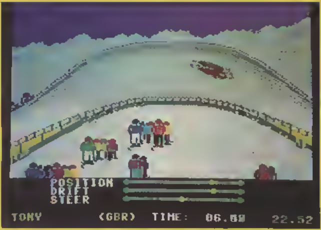
You'll soon find out the meaning of the well known phrase about the "best laid plans" as you now have to perform program as you go for gold Timing is critical as you follow



TONY (GBR) 0:05.3

COMPUTER PACER 0:05.3

n



your program exactly if you are going to win. This is one of the best but most difficult events in the games.

Next, it's your chance to be Eddie "the Eagle" Edwards as you prepare to take your three ski jumps. With a press of a joystick button you begin your descent down the slope, gathering speed before you either soar into the air or fall into the snow leaving your legs dangling. If your takeoff is good you must now concentrate on both style and distance to get the best score before ending the jump with a perfect landing. Well, that's how it's supposed to work.

The slalom is another difficult event along a choice of four courses in which you must weave your way through the flags in the fastest possible time.

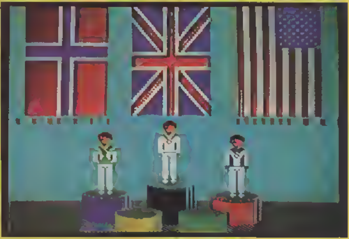
After a gruelling sprint around the 1000, 3000, 5000 or 10,000 metre speed skating rink it's time to grab your skis for the downhill.

Before you launch yourself down the mountain you must position four cameras along the route as these give you the chance to show off your ski stunts. While you're out of camera range you see your view of the mountain disappearing down in front of you as you attempt to steer your way down the course as quickly as possible. The course is marked out with a series of gates but you can cut the occasional corner if you're quick and lucky enough.

When you come into camera range the view switches to a side view where you can throw in some twists and turns to impress the crowd but you should remember that it's the fastest time that matters.

After each event that medals are awarded with the gold medal winner is flanked by the players who came second and third as their flags are lowered behind them.

Whatever the final outcome all will agree that Epyx the Games – Winter Edition is another Epyx epic.



AT A GLANCE

Title: The Games – Winter Edition.
Supplier: Epyx (US Gold), Holford Way, Holford, Birmingham, B6 7AX
Tel: 021 356 3388.
Price: £14.99.
Graphics: Excellent.
Sound: Scraping of skis and crunching of bones.
Playability: A few tricky joystick moves.
Addictiveness: Epyx at its best.

Disk commands

How to get the most out of your drive

By S. Gerton

Most readers of this magazine must be familiar with the 'simple' disk commands of the sort which enable them to scratch files, format disks, initialise the disk and so forth. I put simple in quotes, because Commodore disk commands are among the most obscure I've ever seen on a popular disk operating system.

Be that as it may, there is a whole range of less well-documented commands which let you talk to the disk and drive directly. These are know as Direct Access commands Armed with a knowledge of these, you should be able to get the drive to do what ever you want, short of making breakfast and taking the dog for a walk.

Throughout this feature, bear in mind that we are referring specifically to the 1541, the most common of Commodore drives. Most of the information will be true for other Commodore drives, although there may be minor differences If in doubt, read through your disk manual carefully

Mapping the memory

As you probably know, disk drives are intelligent devices – they contain their own Ram. Table 1 is a memory map of the 1541 Ram, to which you may need to refer.

First of all, buffers All communication between you and the drive will be via buffer As you can see from the table, there are five buffers available. Only four of these, however, are free for use. The fourth buffer is generally reserved for holding an image of the disk's BAM. If you should use SEQ and REL files at the same time, you'll find that Buffer 3 is also not available, because the directory uses it.

In order to read or write information to/from the disk, the sector you wish to operate on must be read into one of the buffers. To use a buffer, it's first necessary to OPEN a channel and specify which buffer. For example:

```
OPEN 1,8,2,"# 2"
```

opens a channel to buffer 2. Good practice, however, dictates not specifying the buffer number, but letting the DOS allocate it. This is done by leaving out the number after the '#' sign:

```
OPEN 1,8,2"# "
```

will have the desired effect.

The INPUT# command can then be used to read in buffer data, but only if the data is alphanumeric and not longer then 88 characters, otherwise you must use the GET command.

GET# doesn't check for null characters, so a trapping line in Basic, or a similar machine code routine is advisable, as in:

```
GET # 2, A$:IF A$="" then A$=CHR$(0)
```

A few more important points to remember are:

* A PRINT# command to the command channel (secondary address 15) will send a Direct Access command to the drive.
* PRINT# to any other channel (s.a. not 15) will send data to one of the buffers.
* INPUT# or GET# to the command channel returns any error messages.
* INPUT# or GET# to any other channel reads data from one of the buffers.

Command formats

There are two ways of entering Direct Access commands. According to taste either

```
PRINT# 15, "U1:" channel-number, drive
```

or

```
PRINT# 15, "U1 channel-number drive"
```

will do the trick.

THE COMMANDS

Block-Read

This command tells the drive to read a disk sector into the open buffer. The command format is "B-R", but can be sndrtened even further to "UI". We'll give an example of how to use it later. Some Commodore drives have a bug in the "B-R" command, so for this reason it always advisable to use the "UI" command

Block-Write

As you might have guessed, this does the exact opposite to Block-Read It takes the contents of the buffer and writes it into the specified sector. The format is "B-W" or "U2" Once again, some drives may have a problem, so it's best to use "U2".

Block-Allocate

This allows the user to reverve blocks on the disk The main use for this is to reserve areas of the disk for special usage. Block-Allocation clears the necessary bits in the BAM after execution of the command. The format is "B-A".

Block-Free

Block-Free does the opposite of Block-Allocate. It sets the bits in the BAM, making specified tracks and sectors available for general use. The format is "B-F".

Block-Execute

Another useful thing you may want to do with blocks is execute them. The Block-Execute (B-E) command is extremely powerful. B-E reads a sector from the disk into a previously opened buffer. The buffer contents are then executed as a machine code program within the buffer

Buffer-Pointer

The Buffer-Pointer (B-P) command tells the DOS the point at which you want to start reading or writing data to/from a buffer

Setting some examples

Let's take a closer look at ndw to use Direct Access commands.

Supposing you wanted to follow a program file through on disk by track and sector without reading any data This can be easily done, because 'link' bytes at the start of each block tell you where the next track and sector are.

This brings us to the first program on the COMMANDS menu, on the disk. If you select 1. PROGRAM TRACE, you'll be able to see the process in action, using the B-R commmand in its UI form.

Let's have a go at using the B-P command Example program 2, READ NAME, reads the diskette name and prints it out. The diskette name starts at character position 144 of track 18 sector 0. It's a simple matter to read this sector into the buffer using B-R

The slow way of getting to the name would be to read all 143 bytes in the buffer until you got to byte 144 Using the Buffer-Pointer command, though, you can get there faster. The example program uses this command to set the pointer to 144, and from there on, it's a simple matter of using GET # to read in. Note in passing that B-R (or UI) automatically sets the pointer to zero every time its executed

Putting it back

Block-Read and Block-Write can often be used together The third program on the menu, CHANGE NAME, is an expanded version of the last, but this time, the disk name read in can be altered in the buffer and then written back to the disk, therefore changing it. Notice that Block-Write does not alter the buffer contents – you have to do this yourself

Allocating space

There is a problem with using the foregoing commands Normally, when you use PRG, SEQ, or REL files, with such commands as SAVE, the BAM is constantly updated as files are written, scratched and so on.

This does not happen with Direct Access commands, meaning that data you have written to disk can be overwritten. Never fear, this can be prevented.

Block-Allocate will reserve space for your data and prevent overwrites If however you try to allocate a block that has already been marked, B-A will give you an Error 65. NO BLOCK, T,S. T and S shows you where the next higher numbered free blocks are

Syntax for Block-Allocate is

8-A drive track sector

For example these lines mark track 16, sector 6 as in use

```
10 OPEN 8,8,15
20 PRINT# 8, "B-A:"0,116,6
```

Of course, you may not want to keep disk areas allocated, so you can deallocate them using the Block-Free (B-F) command. For example, to free the sector we allocated above:

```
10 OPEN 8,8,15
20 PRINT# 8, "B-F:"0;16,6
```

will do the trick

Allocating and freeing blocks only works on blocks that are used on PRG, SEQ and REL files by the DOS. B-W and B-R do not check the BAM before overwriting blocks. With these commands you can write to blocks marked as allocated in the BAM

▶ ▶ ▶ ▶ ▶ ▶ ▶

A common use of this technique is to write a small menu program onto the directory track, track 18. The menu will therefore not take up any extra disk space.

Executing blocks

Block-Execute has a very similar format to the above commands The syntax is:

B-E channel drive track sector

When using B-E it's usual to specify the buffer to be used in the OPEN command, in case the machine code program isn't relocatable.

The following program would read track 16 sector 6 into buffer 2 and execute it.

```
10 OPEN 8,8,15
20 OPEN 4,8,4"# 2"
30 PRINT# 8, "B-E "4,0,16;6
```

Memory commands

Getting access to the disk is only half the story. Memory access commands let you access the disk drive's Ram. The three we shall concern ourselves with are: Memory-Read (M-R); Memory-Write (M-W); and Memory-Execute (M-E).

A good knowledge of the inner workings of the DOS helps when using these commands (see Table 1).

The syntax for the memory read is;

M-R CHR$(l) CHR$(h) CHR$(n)

CHR$(l) is the low byte of the DOS address to be read.

CHR$(h) is the high byte of the DOS address

CHR$(n) is an optional extra parameter that indicates how many bytes are to be read.

Option 4, SHOW BYTES FREE, on the menu shows the use of this command. It shdws hdw to read from disk memory how many bytes are free on the current disk.

Memory-Write does, of course, the opposite to Memory-Read. Writing can only be performed to Dos Ram, page zero, the stack and buffers. You can send a number of bytes to the drive with this command. Syntax is as follows:

M-W CHR$(l) CHR$(h) CHR$(n) CHR$(data1) CHR$(data2) . CHR$(datan)

Memory-Execute calls and executes a program resident in DOS memory The routine must end with an RTS Instruction. Syntax is as follows.

M-E CHR$(l) CHR$(h)

Summary

This feature is only a brief gloss over the possibilities. The list of things you can do with Direct Access commands is endless. The only constraints are your own knowledge and skill

It's worth stressing, before you get going, that you should only play around with these commands using old disks, until you know what you're doing The reasons should be obvious.

Table 1: 1541 Memory Map

DRIVE ADDRESS

HEX	DEC	Description
$0000	0	Command code for buffer 0
$0001	1	Command code for buffer 1
$0002	2	Command code for buffer 2
$0003	3	Command code for buffer 3
$0004	4	Command code for buffer 4
$0006-0007	6-7	Track and sector for buffer 0
$0008-0009	8-9	Track and sector for buffer 1
$000A-000B	10-11	Track and sector for buffer 2
$000C-000D	12-13	Track and sector for buffer 3
$000E-000F	14-15	Track and sector for buffer 4
$0012 0013	18-19	ID for drive 0
$0014-0015	20-21	ID for drive 1
$0016-0017	22 23	ID
$0020-0021	32 33	Flag for head transport
$0030-0031	48-49	Buffer pointer for disk controller
$0039	57	Constant 8 mark for beginning of data block header
$003A	58	Parity for data buffer
$003D	61	Drive number for disk controller
$003F	63	Buffer number for disk controller
$0043	67	Number of sectors per track for formatting
$0047	71	Constant 7 mark for beginning of data block header
$0049	73	Stack pointer
$004A	74	Step pointer for head transport
$0051	81	Actual track number for formatting
$0069	105	Step size for sector division (10)
$006A	106	Number of read attempts (5)
$006F-0070	111 112	Pointer to address for M and B cmds
$0077	119	Device number plus $20 (32) for listen
$0078	120	Device number plus $40 (64) for talk
$0079	121	Flag for listen (I/O)
$007A	122	Flag for talk (I/O)
$007C	124	Flag for ATN from serial bus receiving
$007D	125	Flag for EOI from serial bus
$007F	127	Drive number
$0080	128	Track number
$0081	129	Sector number
$0082	130	Channel number
$0083	131	Secondary address
$0084	132	Secondary address
$0085	133	Data byte
$008B-008D	139-141	Work storage for division
$0094-0095	148-149	Actual buffer pointer
$0099-009A	153 154	Address of buffer 0 $0300
$009B-009C	155 156	Address of buffer 1 $0400
$009D-009E	157 158	Address of buffer 2 $0500
$009F-00A0	159-160	Address of buffer 3 $0600
$00A1-00A2	161 162	Address of buffer 4 $0700
$00A3-00A4	163 164	Pointer to input buffer $0200
$00A5-00A6	165-166	Pointer to buffer error message $02D5
$00B5-00BA	181-186	Record number lo, block number lo
$00BB-00C0	187 192	Record number hi, block number hi
$00C1-00C6	193 198	Write point for REL file
$00C7-00CF	199-204	Record length for REL file
$00D4	212	Pointer in record for REL file
$00D5	213	Side sector number
$00D6	214	Pointer to data block in side sector
$00D7	215	Pointer to record in REL file
$00E7	231	File type
$00F9	249	Buffer number
$0100-0145	256-325	Stack
$0200-0228	512 552	Buffer for command string
$024A	586	File Type
$0258	600	Record length
$0259	601	Track side sector
$025A	602	Sector side sector
$0274	628	Length of input line
$0278	632	Number of filenames
$0297	663	File control method
$0280-0284	640-644	Track of a file
$0285-0289	645-649	Sector of a file
$02D5-02F9	725-761	Buffer for error message
$02FA-02FC	762 764	Number of free blocks
$0300-03FF	768-1023	Buffer 0
$0400-04FF	1024 1279	Buffer 1
$0500-05FF	1280 1535	Buffer 2
$0600-06FF	1536-1791	Buffer 3
$0700-07FF	1792 2047	Buffer 4

High Speed Graphics II

This issue, Allen Webb explains how to move around in his graphics environment

In the first part of this series I described a system for setting up the graphics environment. This time I will describe the first of the routines for transferring information to the display. I have called this the Map module since it is most suited for the displaying of views from above of mazes or landscapes. It does, however, have a range of other uses.

The basic function of the routine is to move a specified rectngular block of information to the screen This data can be a map, the side view of a landscape or whatever you please. This sort of routine is used with scrolling games to update the screen as information is scrolled off. It can, in itself, be used for scrolling although it will only give whole character movement. The routine moves a block of data direct to the screen so the stored information must comprise screen codes and not character strings

So how does it work? The prerequisite is a block of data. Although this is stored as a continuous sequence of values, it represents a rectangular block. The screen is similar since it comprises of 1000 numbers in a row which are displayed as a 40 by 25 character block. In order to handle the data, the routine requires two main chunks of information:

1. The size and location of the data block. This is defined by two parameters, namely the start address(SA) and the width of the data block (WI). Consider any value in this array with coordinates X (horizontal position) and Y (vertical position). The memory location holding the value is then given by:
SA + Y* WI + X
(This assumes that the top left corner of the array has the coordinates X=0 and Y=0).
2. The size of the window on the screen in terms of its position (coordinates of the top left corner) and its size.
Given this information, the data is copied across, byte for byte. The diagram should help to show the relevance of the various parameters.

We also need to consider the colour of the display. Since I didn't want to waste too much memory, I have adopted a short cut. If we limit each character to one colour, we only need a table 256 bytes long As each character is moved to the screen, its colour is also moved to the corresponding position in the colour matrix.

I have included a final feature to extend the system's use in games involving exploration of mazes. In such games you obviously don't want to show areas of the map which haven't been visited. To allow this, a dummy array of the same size and dimensions as the data array is used. Each map location has a corresponding flag in this array. If the flag is zero, the map information is displayed. If the flag is non-zero, a blank (character 160) is displayed.

That's the theory, but how do we use it? The code occupies the block of memory from $OA70 (2672) to $OC92 (3218). The colour table occupies the 256 bytes after the code. There are 7 routines

[illegible]
The routine sets up the information needed [illegible]
[illegible]

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

I recommend that you use the memory behind the Basic Roms and between the Roms for the storage of the data arrays This memory equals about 12k and would allow the storage of a maze with its dummy array occupying six screens in size. If you want to use more memory, you may have to move into Ram used by Basic. If so, you must lower the top of Basic to protect the data. Since this routine sets up the pointers used by the other routines it is vital that you call it first. Failure to do this may cause a crash.

▶ ▶ ▶ ▶ ▶ ▶ ▶

2 SYS 2675,XP,YP,FLAG
This routine moves the map window with its top left corner at XP,YP to the screen. FLAG decides whether a dummy array is used or not. A non-zero value uses the dummy array, a zero value ignores it.

3. SYS 2678,X,Y,BYTE
This routine changes the specified point in the dummy array By using this command to zero the dummy array you can progressively reveal the map as you explore it. The following piece of code shows how to reveal the points adjacent to the player

```
FOR Y = 1 TO 3
FOR X = 1 TO 3
SYS 2678, (XP-2)+X,(YP-2)+Y,0
NEXT X,Y
```

The code assumes that the player is at point XP,YP.

4. SYS 2681,XP,YP
This routine returns the contents of a specified point in the map array in location 900. It can be used to allow movement through the maze and show when objects or nasties are encountered In the next fragment of code, it is assumed that the player is at point XP,YP and is trying to move in direction
DI where:
DI = 1 for north
DI = 2 for east
DI = 3 for south
DI = 4 for west

The code allows the player to move only if the next point in the desired direction is a space (character 32).

```
100 ON DI GOTO 1010,1030,1050,.1070
1010 SYS 2681,XP,YP-1: IF PEEK(900)=32 THEN
YP=YP-1 GOTO 1090
1020 GOTO 1100
1030 SYS 2681,XP+1,YP: IF PEEK(900)=32 THEN
XP=XP+1: GOTO 1090
1040 GOTO 1100
1050 SYS 2681,XP,YP+1· IF PEEK(900)=32 THEN
YP=YP+1: GOTO 1090
1060 GOTO 1100
1070 SYS 2681,XP-1,YP· IF PEEK(900)=32 THEN
XP=XP+1: GOTO 1090
1080 GOTO 1100
1090 . MOVE MAN
1100 ...CAN'T MOVE..
```

5. SYS 2684,XP,YP BYTE
This routine changes the specified point in the map array to the specified value. This allows you to make changes to the map as play progresses. You could, for example, generate random earthquakes to block or reveal passages. A crude way of doing this would be

```
100 FOR Z = 1 TO 20
110 CH=160: IF RND(1)<.5 THEN CH=32
120 SYS 2684,WI*RND(1),HI*RND(1),CH
130 NEXT
```

This example changes 20 locations to either a wall or a passage. It assumes a map of width WI and height HI It does not check to see what the status of each point is before it is changed.

6. SYS 2678,CHAR,COL
This changes the colour of the specified character in the colour table

7 SYS 2690,XP,YP
This final routine returns the value of the specified point in the dummy array in location 901. This, in conjunction with routine 3 allows you to use the dummy array for other jobs. Imagine that you write a game with a 3D view of the passage but with no map and you give the option of marking the locations visited with a piece of chalk. You could identify visited locations by setting their values in the dummy array to special values.

Due to the way the routines are written, the only limitation to the number of display windows you have is memory for data storage. All you need to do is call the initialising routine before you display each window. You could therefore show two parts of the maze at once or show both top and side views of the scene.

If used simply as a display routine, there is no limitation to what information you can store. It can, however, use a data array in conjunction with other display routines given in this series. As I will describe later, this may impose certain constraints. See you next issue

Loading the program

[illegible]

COMMODORE - 64
SOFTWARE ON DISK

Easyscript	£14 95	Easyspell	£9 95
Future Finance	£12 95	Simon s Basic Extension	£9 95
Logo	£12.95	Pilot	£9.95
Zoom Pascal	£9 95	Ultrabasic 64	£9 95
Pascal 64	*£12.95	Ada	£12.95
Assembler/Monitor 64	*£12 95	Assembler tutor	£9 95
* Both for	£19 95	Programmer's Utilities	£9 95
Basic 64 Compiler	£9 95	Calc Result Advanced	£14 95
DMF Database	£9 95	The Real You	£4 95
Easyfile	£12.95	Bradon s Basic	£5.95
Money Manager	£9 95	Spirit of The Stones	£4.95

COMMODORE-64 GAMES ON DISK - £3 95 each or any 3 for £10

Zork I *
Zork II *
Zork III *
Suspend *
Starcross *
Deadline
Pitfall
Pitfall II
Beamrider
Hero
Zenji
Pastfinder
Mindshadow
Empire
* Also available for Plus/4

Web Diemension
Road Race
Footdaller of The Year
Hopeless
Inhentance
Mission Elevator
India OC
Alley Kat
Tass Times
Dandy
Spindizzy
Exploror
Prodigy
Ankh

Chamelion
Championship Wrestling
The Vikings
Galactic Controller
Unorthodox Engineers
Little Computer People
Master of The Lamps
Golf Construction Set
Leviathan
Bugsy
Dragon s Lair
Call Defense
Arcade Extravaganza
Druid
High Flyer

Some quantities limited State alternatives where possible

BOOKS:

Anatomy of 1541 Disk drive	£4.95
1541 Repair & Maintenance	£12.95
C64 Programmers Ref Guide	£5.95
Anatomy of 1571 Disk Drive	£9.95

Commodore 1581 Disk Drive:

800K formatted capacity using 3 5 disks Our price £184 95 inc 5 blank disks

Prices include p&p in U K. Add £1 per item overseas Access & Visa accepted COD orders add £1 50 Send SAE for complete list or further details on any item TEL: 0706 524304

ADAMSOFT 18 Norwich Avenue, Rochdale, Lancs.
OL11 5JZ

PRINT ENHANCER Plus/4 disk £19.99

Plus/4 desktop publishing. Proportionally spaced, high quality printouts, from BASIC 3+1 and script/plus Supplied with 15 fonts and font designer program MPS 801/803 VIC 1525

RS232 INTERFACE 64, 128 or Plus/4* ...£34.99

Measuring only 12x69x57mm, the smallest and neatest unit available. Conforms to Commodore 1011A user-port standard Compatible with BASIC, Easyscript, Superscript Superbase, Mini Office, Script Plus etc. Connect to RS232 printers modems and other devices Supplied with 1 metre of cable (add 75p extra metre') terminated with a male (female*) 25 way D-connector, or a custom cable made to your requirements Our after sales service for unusual printers guarantees to get you up and running Supplied with a terminal emulator program, file transfer and other utilities on tape/disk'

IBM PC File Transfer Utility £9.99

Captures text or binary files via the COMI RS232 port. Supplied on a 360k IBM format 5¼" disk Simple instructions provided by on-line help

VIEWDATA TERMINAL 64 or Plus/4 ....£14.99

Access Prestel Micronet, CityService etc. Works with any user-port RS232 interface and 1200/75 modem

CENTRONICS CABLE 64, 128 ....£18.99

Use with Easyscript Superscript etc Driver for BASIC on disk or tape *

6502 ASSEMBLER 64 or Plus/4* tape/disk £12.99

A sophisticated two-pass symbolic assembler and text editor which supports tape and disk filing The assembler, editor source and object code may all be resident in memory simultaneously, facilitating rapid and interactive code development Very fast edit/Assembler/test cycles

Z80 EMULATOR/ASSEMBLER 64 disk £12.99

A unique integrated Z80 development package The emulator compiles Z80 source code into optimised 6502 which runs on the 64 at about one sixth the speed of a 2MHz Z80 The cross-assembler generates hex or binary Z80 object files The disk contains a powerful editor and example programs

Please specify' options. Send cheque/PO. or order by access. Prices include VAT and P&P. Overseas orders add £2.50. Allow up to 1 week for delivery

YORK ELECTRONIC RESEARCH

The Fishergate Centre, Dept YC, 4 Fishergate, York YO1 4AB
Tel (0904) 610722

COMING SOON - CBM 64. (DISC) £14.95, ATARI ST/AMIGA £19.95. AMSTRAD £9.95, IBM - P.O.A

From the Designer of 'The Double' comes .

TRACK SUIT MANAGER

The first ever football game with real match tactics.

REVIEWS SAY

Commodore User says . The match is the best I've ever seen TSM is the most playable game of its genre One of the most frustrating fun, compelling, exciting and addictive games I've ever played 8/10

C & V Games says ... The best ever football managerial game ever in the history of the world no messing straight up A very impressive start for Goliath 9/10

Zzap says ..
Presentation 96% Lastability 86%
Hookability 93% Overall 89%
Easily the best of its type

Sinclair User says . The best ever managerial game the one your speccy has been waiting for 93%

Your Sinclair says . . I'm amazed at the one Some day all footie games will be made this way If you don't like TSM your dead from the neck up 8/10

Your Commodore says TSM is one of the best games I've ever played and certainly highlights the problems of a national team manager

WE SAY

Sophisticated Match Play with Adjustable Team Formations ✶ Free Kicks ✶ Throw Ins ✶ Offsides ✶ Individual Player Abilities ✶ Goal Kicks ✶ Corners ✶ Penalties ✶ Tackles ✶ Fouls ✶ Select Player for Set Pieces ✶ Injuries ✶ Bookings ✶ Sending Off's ✶ Suspensions ✶ Substitutes ✶ Heading ✶ Genuine Penalty Shoot Out in Quarter Semi & Final Matches ✶ Extra Time ✶

Individual Player & Team (Defence, Midfield, Attack) tactics Defensive ✶ Attacking ✶ Normal ✶ Quick Counter Attacks ✶ Zonal Marking ✶

GOLIATH GAMES is a 'NEW' Software House specialising in ... ADVANCED . . INVOLVED ... ORIGINAL SOFTWARE.

Available from all good computer retailers.
If you have any difficulty then
Including V A T Plus 50p post & packing

GOLIATH GAMES
The GIANTS amongst software

PLEASE SEND CHEQUES/P.O. TO:-
48 LOCKING ROAD
WESTON-SUPER-MARE
AVON BS23 3DN
0934 622538

COMMODORE 64, SPECTRUM

DON'T GET LEFT OUT... GET IN ON THE ACTION

COMMODORE DISK USER is a lot more than just another computer magazine. Every issue carries a diskette containing more than £30 worth of software ranging from serious programming utilities to arcade games. There are plenty of Commodore magazines on the market, but we believe that this is the *first* to cater for disk users of all ages and tastes.

COMMODORE DISK USER is what you have been waiting for – take out a subscription TODAY!

Please begin my subscription(s) to COMMODORE DISK USER with the .... issue

I enclose my cheque/money order for £ .... made payable to Argus Specialist Publications Ltd

or debit £ .... from my Access/Barclaycard No ....

valid from .... to ....

NAME (Mr/Mrs/Miss) ....

ADDRESS ....

....

Postcode ....

Signature ....

Date ....

Please use BLOCK CAPITALS and use post codes

SUBSCRIPTION RATES

£15.00 for 6 issues U.K.
£18.00 for 6 issues EUROPE
£18.20 for 6 issues MIDDLE EAST
£19.30 for 6 issues FAR EAST
£18.40 for 6 issues REST OF WORLD
Airmail Subscription Rates on Request

Send your remittance to:
INFONET LTD., 5 River Park Estate,
Berkhamsted, Herts. HP4 1HL.